actuators

Article

Neural Adaptive Robust Motion-Tracking Control for Robotic Manipulator Systems

Daniel Galvan-Perez [1], Hugo Yañez-Badillo [2], Francisco Beltran-Carbajal [3,*], Ivan Rivas-Cambero [1], Antonio Favela-Contreras [4] and Ruben Tapia-Olvera [5]

[1] Departamento de Posgrado, Universidad Politécnica de Tulancingo, Tulancingo de Bravo 43629, Hidalgo, Mexico
[2] Departamento de Investigación, Tecnológico de Estudios Superiores de Tianguistenco, Santiago Tianguistenco 52650, Estado de México, Mexico
[3] Departamento de Energía, Unidad Azcapotzalco, Universidad Autónoma Metropolitana, Av. San Pablo No. 180, Col. Reynosa Tamaulipas, Mexico City 02200, Mexico
[4] Tecnologico de Monterrey, School of Engineering and Sciences, Ave. Eugenio Garza Sada 2501, Monterrey 64849, Nuevo León, Mexico
[5] Departamento de Energía Eléctrica, Universidad Nacional Autónoma de México, Mexico City 04510, Mexico
* Correspondence: fbeltran@azc.uam.mx

Abstract: This paper deals with the motion trajectory tracking control problem based on output feedback and artificial neural networks for anthropomorphic manipulator robots under disturbed operating scenarios. This class of manipulator robots constitutes nonlinear dynamic systems subjected to disturbance torques induced mainly by work payload. Parametric uncertainty and possible dynamic modeling errors stand for other kind of disturbances that can deteriorate the efficiency and robustness of the tracking of controlled nonlinear robotic system trajectories. In fact, the presence of unknown dynamic disturbances is unavoidable in industrial robotic engineering systems. Therefore, for high-precision applications, such as laser cutting, marking, or welding, effective control schemes should be designed to guarantee adequate motion profile tracking planned on this class of disturbed nonlinear robotic system. In this context, a new adaptive robust motion trajectory tracking control scheme based on output feedback and artificial neural networks of anthropomorphic manipulator robots is presented. Three-layer B-spline artificial neural networks and time-series modeling are properly exploited in the design of novel adaptive robust motion tracking controllers for robotic applications of laser manufacturing. In this way, dependency on detailed nonlinear mathematical modeling of robotic systems is considerably reduced, and real-time estimation of uncertain dynamic disturbances is not required. Furthermore, several cases studies to demonstrate the motion planning tracking control robustness for a class of MIMO nonlinear robotic systems are described. blue Insights for the extension of the introduced output-feedback adaptive neural control design approach for other architecture of nonlinear robotic systems are depicted.

Keywords: active disturbance rejection; artificial neural networks; laser manufacturing; manipulator robot; trajectory tracking control

check for updates

Citation: 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. https://doi.org/10.3390/act11090255

Academic Editor: Haim Abramovich

Received: 19 August 2022
Accepted: 5 September 2022
Published: 7 September 2022

1. Introduction

The study and application of intelligent control techniques, such as fuzzy logic and neural networks, have gained widespread recognition in recent years. The basis of these techniques consists of learning in a prescribed manner the input–output behavior of a system to subsequently be controlled [1]. The importance of these types of techniques is found in nonlinear systems, variants in time, and those that are subjected to different types of disturbances [2]. Manipulator robots have complex nonlinear dynamics that can make accurate and robust control difficult. In today's manufacturing environments, velocity and accuracy requirements increase when compared to earlier generations of robots. However, the efficiency of fixed-gain PID controllers often is limited for providing adequate performance in real-time operations of robot motion [3].

Since there are no adaptive or learning capabilities, control accuracy is significantly affected when a variation due to unknown frictions and disturbing torques is presented during motion control in laser manufacturing applications, such as cutting, welding, marking, and additive 3D printing [4]. The main purpose of controlling a manipulator robot is to accomplish a specific task, such as payload transport [5], and motion profile tracking for material processing [6], among others. To perform the required tasks, the manipulator is commanded for reaching the desired position, velocity, acceleration, or force at a specific time [7]. The nonlinear nature of manipulator robots makes it difficult to obtain an adequate control law to perform different programmed tasks. There are computational methods that improve the control laws efficiency, such as the artificial neural networks [8] that are used to online update the control parameters value to improve the robotic system's closed-loop response. The exhibited adaptive behavior learns from the nonlinear nature of the robot intelligently. So using neural networks for the tuning of the parameters of a controller applied to a nonlinear and strongly coupled system represents a great advantage. The recent renewed interest in neural networks can be attributed to different factors, one of them is that different learning techniques have been developed for sophisticated artificial neural network architectures, which are capable of eliminating the limitations of the past associated with simple neural networks [9]. Neural networks are parallel computer models that mimic a human's learning process. The networks process information by adapting their connections and nodes, based on the information they are given [10]. One important feature is the intrinsic parallel architecture that allows fast computation of the solution. This is achieved when the networks are implemented on digital computers or in specialized hardware [11].

Artificial neural networks (ANN) have been used for the control of manipulator robots for different applications in recent years. In [12], a trajectory planning methodology of a manipulator robot is proposed through the design of an adaptive robust controller that uses a neural network activated with radial basis functions; this is done to solve nonlinearity and uncertainty problems and improve the performance during trajectory tracking. In [13], a fixed-time control algorithm for the trajectory tracking of manipulator robots with uncertainty and input saturation is designed, which combines the non-singular terminal sliding mode control with the reinforcement learning method. To compensate for the saturation of the actuators due to the effect of joint torque overflow, a nonlinear anti-windup compensator is designed. Authors in [14] introduce a multi-layer neural network with iterative learning for controller design to compensate for the dynamics of an articulated industrial manipulator robot for trajectory tracking tasks. The iterative learning control commands the robot motion, which corresponds to a set of desired joint trajectories. Additionally, the torques of the input/output trajectories are used for training the inner control loop through inverse dynamics. In this scheme, the trajectory is used as the input desired joint of the robot, and the output is related to the commanded robot motion. In [15], a modified artificial neural network algorithm is suggested to be applied as an adaptive adjustment algorithm. A mathematical modulation is introduced to promote the scanning form, and no initial parameters are set. The suggested algorithm is tested to select the gains, and an appropriate configuration is selected. This algorithm is applied for improving a PID-like joint motion controller for a manipulator robot. In [16], the output error constraints and input saturation of robot manipulators are considered and presented with a trajectory tracking controller to attend to these requirements simultaneously through an artificial neural network structure with radial basis functions that can approximate the pooled uncertainties. In [17], a deep convolutional neural network is designed to implement a fractional order sliding mode control strategy for trajectory tracking control in manipulator robots. The neural network compensates for system uncertainty without the need to know the upper bounds, thus the controller switching gain is greatly reduced. The main advantage of the controller used lies in the elimination of chattering, so continuous control signals can be obtained and applied in practical applications. In [18], a control scheme is proposed to handle the unknown dynamics and the external disturbances in a manipulator robot by combining a self-tuning PID structure with an artificial neural network using radial basis activation functions to handle the different uncertainties. In [19], to track the trajectory of a robot and supply constant force during contact working conditions, a PID fuzzy neural network controller is developed. For manufacturing applications, changing the contact force during machining operations will affect the quality, so it is crucial to maintain them constantly throughout the process. For the parameters of the PID controller to be adjusted quickly and effectively, a fuzzy algorithm is proposed that includes neural networks in such a way that the controller has two functions: path following and bounded shifting online whilst maintaining the contact force constant. In [20], an adaptive neural control scheme for manipulator robots that includes actuator dynamics under model uncertainty is proposed based on prescribed performance without measuring input current, acceleration, or velocity. A neural adaptive second-order PID controller is combined with an acceleration velocity observer. The proposed method demonstrates that semi-global bounds are found for tracking and state observation errors, and they eventually converge in a small circle about the origin.

With the B-spline artificial neural networks (Bs-ANN), adaptive control and modeling of systems are possible both online and with nonlinearities taken into account [21]. In the literature, the estimation of harmonics and DC offset component of multi-frequency oscillating electric signals [22], control of electric power systems [23], different types of electric motors, such as DC shunt [24], induction [25] and permanent magnet synchronous [9] as well as unmanned aerial vehicles [26] are some examples of the successful implementation of B-spline neural networks. Nevertheless, there is no evidence of the previous implementation in anthropomorphic manipulator robots. The main advantage of the B-spline neural network is that it can limit the input to a set of values defined in the base function. This reduces computational effort and time because there are only a limited number of basis functions involved in the output; not all weights need to be calculated every time iterations are performed [26]. The proposal allows calculating and updating the values of the parameters of the proposed controller through the neural network for each operating condition of the robot. A proper design requires prior information: the maximum and minimum values for the input signal and defining the shape of the basis functions. This information allows for limiting the input space of the B-spline neural networks, which improves their convergence and stability in adaptation. Additionally, this information is used to determine the optimal weights for the neural network [27]. The B-spline neural networks used in this work are created by third-order uni-variable basis functions considering that the input signal is bounded.

The main contribution of this work is a new neural adaptive robust motion trajectory tracking control technique based on output feedback for a very important class of nonlinear robotic manipulator systems used in applications of laser manufacturing. In this context, advantages and differences with respect to other relevant contributions reported in the literature are as follows. Position measurements are only required. Time derivatives of measured output signals are not requested. Accurate real-time estimation of different types of disturbances is unnecessary. B-spline artificial neural networks are used correctly to prevent high-gain control actions to suppress unknown time-varying disturbances. The number of control parameters that need to be adaptively adjusted is lessened. Dependency on detailed nonlinear mathematical modeling of the manipulator robot system is reduced. Robustness against external disturbances, such as external vibrating torques, is also obtained. Moreover, the introduced output feedback adaptive neural control design approach can be extended for other architectures of complex nonlinear robotic systems.

A summary of the contents of the manuscript is as follows. The robotic system model is described in Section 2. The robot motion control approach is presented in Section 3. In Section 4, the artificial neural networks-based robot motion control is described. Subsequently, some numeric simulation experiments are presented in Section 5 to highlight the performance of the introduced methodology. Finally, some conclusions and remarks are mentioned in the conclusions section.

2. 3-DOF Robot Mathematical Modeling

2.1. System Description

The geometry of interest in this work is anthropomorphic as shown in Figure 1. The robotic system configuration is composed of three rotational joints. This configuration represents the most dexterous structure, which can be adapted to almost any type of task.

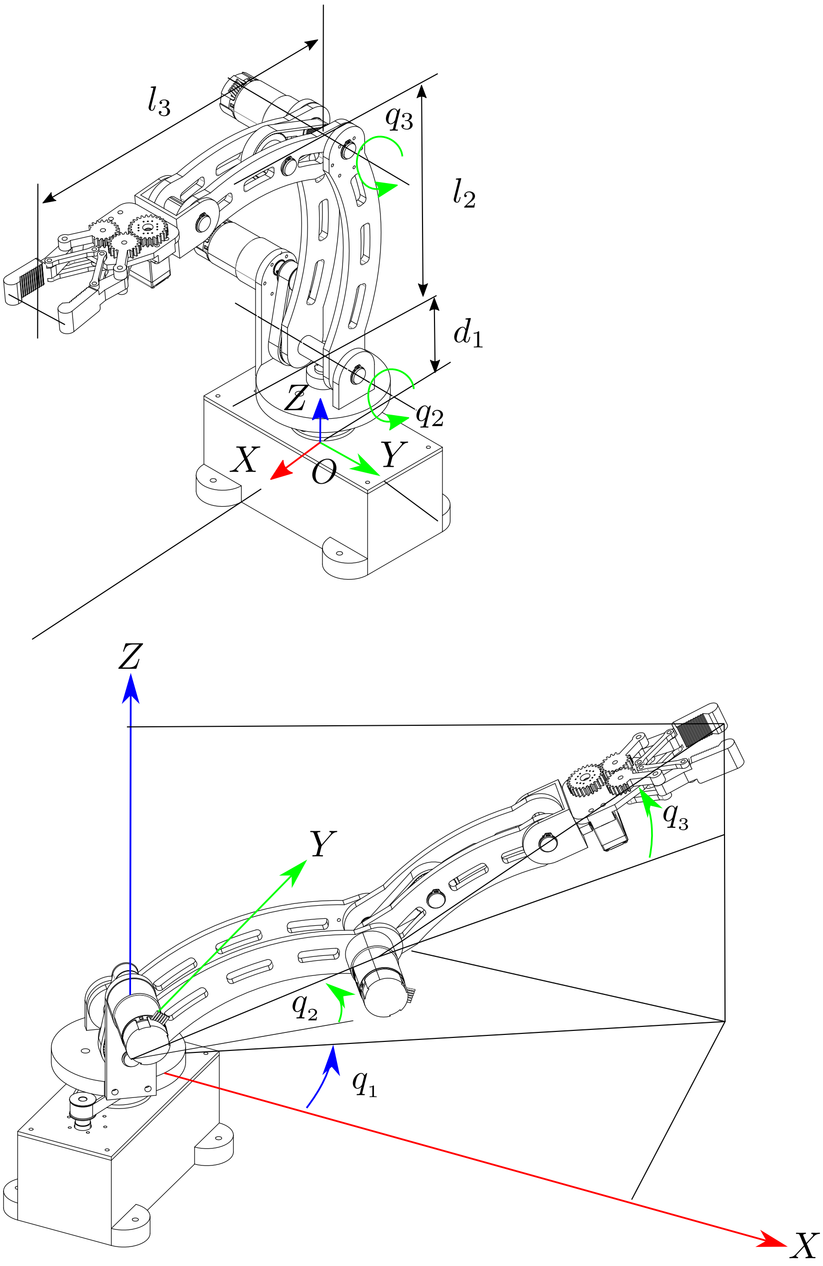


Figure 1. Architecture of the anthropomorphic robotic system.

Figure 2 depicts projections of the anthropomorphic manipulator robot on XY and XZ planes for the derivation of a nonlinear dynamic model. Generalized coordinates of the configuration space are denoted by q_i, $i = 1, \ldots, 3$. Here, q_1 is the rotation angle of link 1 about the Z axis. q_2 and q_3 are the rotation angles of links 2 and 3 about the Y axis. d_1 is the distance between point O of the fixed inertial coordinate reference frame XYZ and the rotation axis of q_2. l_2 corresponds to the distance between the rotation axis of q_2 and the rotation axis of q_3. l_{c_2} is the distance from the rotation axis of q_2 to the center of mass of link 2. l_3 is the distance between the rotation axis of q_3 and the center of the end effector. l_{c_3} is the distance from the rotation axis of q_3 to the center of mass of link 3. m_1 is the mass of link 1. m_2 corresponds to the mass of link 2. m_3 is the mass of link 3, which also includes the mass of the end effector.

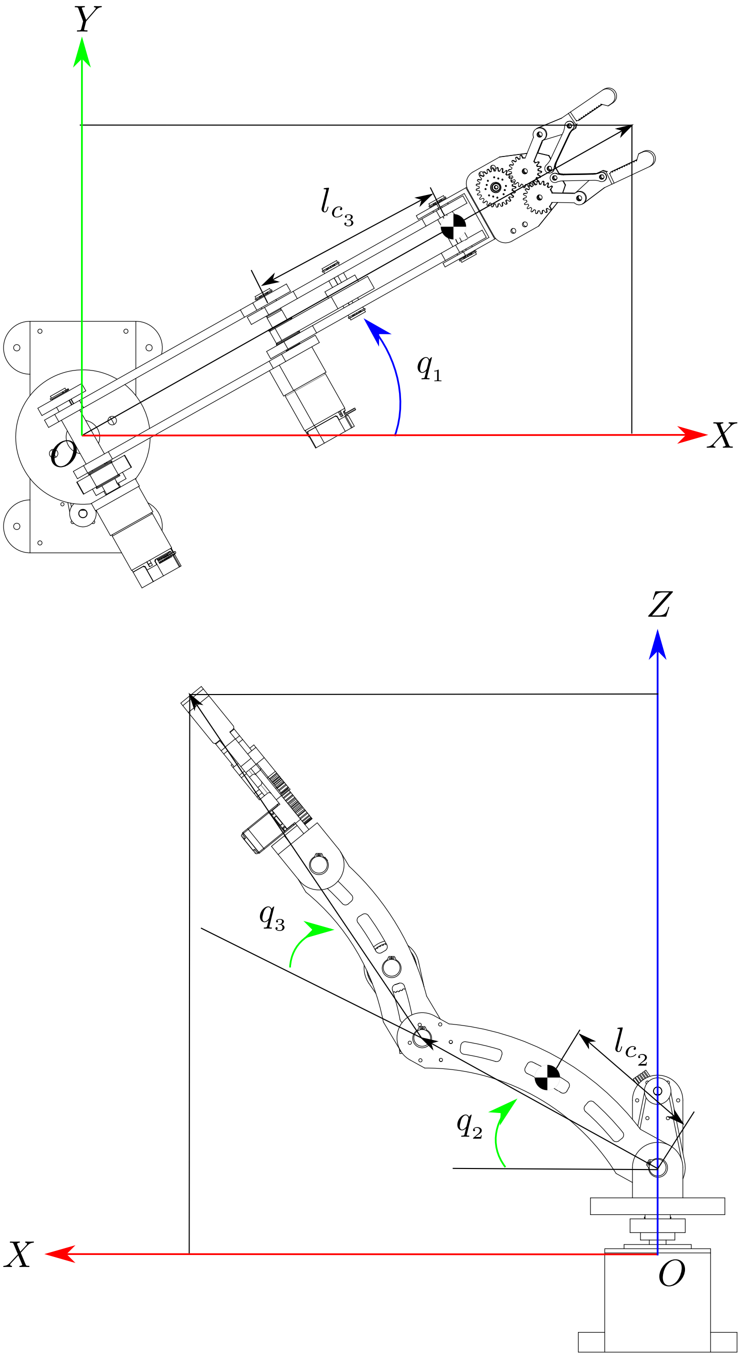


Figure 2. Projections on XY and XZ planes for mathematical modeling purposes.

2.2. Kinematic Modeling

The direct kinematic model of a 3-DOF anthropomorphic manipulator robot [28,29] is expressed as

$$\begin{aligned} x &= \cos(q_1)[l_2\cos(q_2) + l_3\cos(q_2+q_3)] \\ y &= \sin(q_1)[l_2\cos(q_2) + l_3\cos(q_2+q_3)] \\ z &= d_1 + l_2\sin(q_2) + l_3\sin(q_2+q_3) \end{aligned} \tag{1}$$

The inverse kinematic model of the 3-DOF anthropomorphic manipulator robot [28,29] is then described as

$$\begin{aligned} q_1 &= \tan^{-1}\left(\frac{y}{x}\right) \\ q_2 &= \tan^{-1}\left(\frac{z-d_1}{\sqrt{x^2+y^2}}\right) - \tan^{-1}\left[\frac{l_3\sqrt{1-\left(\frac{x^2+y^2+(z-d_1)^2-l_2^2-l_3^2}{2l_2l_3}\right)^2}}{l_2+l_3\left(\frac{x^2+y^2+(z-d_1)^2-l_2^2-l_3^2}{2l_2l_3}\right)}\right] \\ q_3 &= \tan^{-1}\left[\frac{\pm\sqrt{1-\left(\frac{x^2+y^2+(z-d_1)^2-l_2^2-l_3^2}{2l_2l_3}\right)^2}}{\left(\frac{x^2+y^2+(z-d_1)^2-l_2^2-l_3^2}{2l_2l_3}\right)}\right] \end{aligned} \tag{2}$$

2.3. Dynamic Modeling

The analysis of a manipulator robot requires the use of differential equations to understand the relationship between the elements that compose it. By modeling the dynamic behavior of a manipulator robot, all physical phenomena found in its mechanical structure are explained, including inertial effects, centripetal and Coriolis forces, gravitational torque, and friction, which are physical phenomena resulting from the robot's dynamic nature [30]. The development of the dynamic model of an anthropomorphic manipulator robot through the Euler–Lagrange methodology was widely studied in [28–30].

The nonlinear behavior of a manipulator robot can be thus modeled by the Euler–Lagrange equations

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \quad i = 1,\ldots,n \tag{3}$$

with

$$L = K - V \tag{4}$$

where L is the Lagrangian, K and V are the kinetic and potential energy. n denotes the number of degrees of freedom, $q = (q_1, q_2, \ldots, q_n)$ represents the generalized coordinates of the robotic system, and Q_i stands for components of the generalized force.

The kinetic energy of the robotic system depicted in Figures 1 and 2 is given by

$$K = K_1 + K_2 + K_3 \tag{5}$$

with

$$\begin{aligned}
K_1 &= \frac{1}{2} I_{z_1} \dot{q}_1^2 \\
K_2 &= \frac{1}{2} m_2 l_{c_2}^2 [(1 - \sin(q_2)^2) \dot{q}_1^2 + \dot{q}_2^2] + \frac{1}{2} I_{z_2} \dot{q}_1^2 + \frac{1}{2} I_{y_2} \dot{q}_2^2 \\
K_3 &= \frac{1}{2} m_3 [(l_2^2 + l_2^2 \cos(2q_2) + 2 l_2 l_{c_3} \cos(2q_2 + q_3) + 2 l_2 l_{c_3} \cos(q_3) + l_{c_3}^2 \cos(2q_2 + 2q_3) \\
&\quad + l_{c_3}^2) \frac{\dot{q}_1^2}{2} + (l_2^2 + 2 l_2 l_{c_3} \cos(q_3) + l_{c_3}^2) \dot{q}_2^2 + (2 l_2 l_{c_3} \cos(q_3) + 2 l_{c_3}^2) \dot{q}_2 \dot{q}_3 + l_{c_3}^2 \dot{q}_3^2] + \frac{1}{2} I_{z_3} \dot{q}_1^2 \\
&\quad + \frac{1}{2} I_{y_3} (\dot{q}_2 + \dot{q}_3)^2
\end{aligned}$$

The potential energy is obtained as

$$V = V_1 + V_2 + V_3 \tag{6}$$

with

$$\begin{aligned}
V_1 &= m_1 g d_1 \\
V_2 &= m_2 g [d_1 + l_{c_2} \sin(q_2)] \\
V_3 &= m_3 g [d_1 + l_2 \sin(q_2) + l_{c_3} \sin(q_2 + q_3)]
\end{aligned}$$

From Equations (3) and (4), nonlinear dynamics of the robotic system can be then described by the vector differentialequation

$$\mathbf{D}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} - \boldsymbol{\tau}_d \tag{7}$$

where $\mathbf{D}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ stands for the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ represents the Coriolis and centripetal matrix and $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^n$ denotes the torque vector due to gravity [28]. Moreover, $\boldsymbol{\tau}$ and $\boldsymbol{\tau}_d \in \mathbb{R}^n$ are the joint driving torque and the external load torque, respectively.

From Equation (7) and without loss of generality, the dynamic model for a 3-DOF anthropomorphic manipulator robot is given by

$$\begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} g_{11} \\ g_{21} \\ g_{31} \end{bmatrix} g = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} - \begin{bmatrix} \tau_{d_1} \\ \tau_{d_2} \\ \tau_{d_3} \end{bmatrix} \tag{8}$$

Grouping terms according to the joint accelerations of the robot $(\ddot{q}_1, \ddot{q}_2, \ddot{q}_3)$, the inertia matrix is expressed as

$$\mathbf{D}(\mathbf{q}) = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & d_{23} \\ 0_{31} & d_{32} & d_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} \tag{9}$$

with

$$\begin{aligned}
d_{11} &= I_{z_1} + I_{z_2} + I_{z_3} + \left(\frac{m_2}{2} \right) [l_{c_2}^2 (1 + \cos(2q_2))] \\
&\quad + \left(\frac{m_3}{2} \right) [2 l_2 l_{c_3} (\cos(2q_2 + q_3) + \cos(q_3)) + l_2^2 (1 + \cos(2q_2)) + l_{c_3}^2 (1 + \cos(2q_2 + 2q_3))] \\
d_{22} &= I_{y_2} + I_{y_3} + m_2 l_{c_2}^2 + 2 m_3 l_2 l_{c_3} \cos(q_3) + m_3 \left(l_2^2 + l_{c_3}^2 \right) \\
d_{23} &= I_{y_3} + m_3 l_2 l_{c_3} \cos(q_3) + m_3 l_{c_3}^2 \\
d_{32} &= I_{y_3} + m_3 l_2 l_{c_3} \cos(q_3) + m_3 l_{c_3}^2 \\
d_{33} &= I_{y_3} + m_3 l_{c_3}^2
\end{aligned}$$

Similarly, by grouping terms according to the joint velocities of the robot $(\dot{q}_1, \dot{q}_2, \dot{q}_3)$, the matrix of centripetal and Coriolis forces is expressed as

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 0 & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \tag{10}$$

with

$$\begin{aligned}
c_{12} &= -\left(m_2 l_{c_2}^2 \sin(2q_2) + m_3\left(l_2^2 \sin(2q_2) + 2 l_2 l_{c_3} \sin(2q_2 + q_3) + l_{c_3}^2 \sin(2q_2 + 2q_3)\right)\right)\dot{q}_1 \\
c_{13} &= -(m_3 l_{c_3}(l_2(\sin(2q_2 + q_3) + \sin(q_3)) + l_{c_3} \sin(2q_2 + 2q_3)))\dot{q}_1 \\
c_{21} &= \left(\left(\frac{m_2}{2}\right) l_{c_2}^2 \sin(2q_2) + \left(\frac{m_3}{2}\right)\left(l_2^2 \sin(2q_2) + 2 l_2 l_{c_3} \sin(2q_2 + q_3) + l_{c_3}^2 \sin(2q_2 + 2q_3)\right)\right)\dot{q}_1 \\
c_{22} &= -2 m_3 l_2 l_{c_3} \sin(q_3)\dot{q}_3 \\
c_{23} &= -m_3 l_2 l_{c_3} \sin(q_3)\dot{q}_3 \\
c_{31} &= \left(\frac{m_3}{2}\right)\left(l_{c_3}^2 \sin(2q_2 + 2q_3) + l_2 l_{c_3}(\sin(2q_2 + q_3) + \sin(q_3))\right)\dot{q}_1 \\
c_{32} &= m_3 l_2 l_{c_3} \sin(q_3)\dot{q}_2
\end{aligned}$$

The gravitational torque vector of the manipulator robot is finally expressed as

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix} 0 \\ (m_2 l_{c_2} + m_3 l_2)\cos(q_2) + m_3 l_{c_3} \cos(q_2 + q_3) \\ m_3 l_{c_3} \cos(q_2 + q_3) \end{bmatrix} g \tag{11}$$

3. Robot Motion Control Synthesis

The mathematical model (7) has been widely used to describe the controlled dynamics of several architectures of complex nonlinear robotic systems [28,29].From Equation (7), the tracking error dynamics on controlled nonlinear robotic system trajectories can be expressed as follows:

$$\ddot{\mathbf{e}}_{\mathbf{q}} = \mathbf{v} + \boldsymbol{\xi} \tag{12}$$

with

$$\boldsymbol{\xi} = -\ddot{\mathbf{q}}^{\star} - \mathbf{D}^{-1}(\mathbf{q})\mathbf{G}(\mathbf{q}) - \mathbf{D}^{-1}(\mathbf{q})\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{D}^{-1}(\mathbf{q})\boldsymbol{\tau}_d \tag{13}$$

where $\mathbf{v} = \mathbf{D}^{-1}(\mathbf{q})\boldsymbol{\tau}$ denotes an auxiliary (virtual) control input vector.

The trajectory tracking error vector is $\mathbf{e}_{\mathbf{q}} = \left[e_{q1}\ e_{q2}\ e_{q3}\right]^{\mathsf{T}}$ with $e_{q_i} = q_i - q_i^{\star}$ for $i = 1, 2, 3$. Here, $q_i^{\star}(t)$ stands for angular position reference trajectories planned for the operation of the robotic system.

In the present study, the integral reconstruction approach of velocity state variables is suitably exploited [31]. In this fashion measurements of velocity signals are unnecessary. Dependence on disturbed nonlinear dynamic system models is also reduced. Furthermore, differentiation of generalized coordinates with respect to time is not requested, avoiding the possible generation of undesirable corrupting noise during the measurement signal processing. Thus, from Equation (12), the integral reconstructor of the reference trajectory tracking error velocity vector is proposed as follows:

$$\widehat{\dot{\mathbf{e}}}_{\mathbf{q}} = \int_{t_0}^{t} \mathbf{v}\, dt \tag{14}$$

For robust control design purposes, $\boldsymbol{\xi} = [\xi_1\ \xi_2\ \xi_3]^\mathsf{T}$ is considered a completely unknown time-varying disturbance vector to be actively rejected, which can be approximated into a self-adjusting, very small window of time by the Taylor series polynomial expansion

$$\boldsymbol{\xi} = \mathbf{p}_0 + \mathbf{p}_1 t \tag{15}$$

Moreover, parameter vectors of the disturbance signal model (15), $\mathbf{p}_0 = \left[p_{0_1}\ p_{0_2}\ p_{0_3}\right]^\mathsf{T}$ and $\mathbf{p}_1 = \left[p_{1_1}\ p_{1_2}\ p_{1_3}\right]^\mathsf{T}$, are assumed to be unknown as well. Small uncertainties in components of the matrix $\mathbf{D}(\mathbf{q})$ and dynamic modeling errors could be also considered into the disturbance vector $\boldsymbol{\xi}$. In contrast to other active disturbance rejection control approaches, the real-time estimation of disturbances and time derivatives of position output signals is not required in the present contribution.

Then, the real and reconstructed tracking error velocity vectors holds the following relationship:

$$\dot{\mathbf{e}}_\mathbf{q} = \widehat{\dot{\mathbf{e}}}_\mathbf{q} + \mathbf{a}_0 t + \mathbf{a}_1 t^2 \tag{16}$$

where vector parameters $\mathbf{a}_0 = \left[a_{0_1}\ a_{0_2}\ a_{0_3}\right]^\mathsf{T}$ and $\mathbf{a}_1 = \left[a_{1_1}\ a_{1_2}\ a_{1_3}\right]^\mathsf{T}$ are also assumed to be unknown, which depend on initial conditions of the nonlinear robotic system and parameters of the polynomial signal model (15). The interested reader on integral reconstructors as an alternative to bypass time derivatives of measurement signals and design of asymptotic state observers is referred to the contribution [31].

The auxiliary control vector $\mathbf{v}$ is proposed as follows

$$\mathbf{v} = -\mathbf{B_4}\widehat{\dot{\mathbf{e}}}_\mathbf{q} - \mathbf{B_3}\mathbf{e}_\mathbf{q} - \mathbf{B_2}\int \mathbf{e}_\mathbf{q} - \mathbf{B_1}\int^{(2)} \mathbf{e}_\mathbf{q} - \mathbf{B_0}\int^{(3)} \mathbf{e}_\mathbf{q} \tag{17}$$

where

$$\mathbf{B}_i = \begin{bmatrix} \beta_{i_1} & 0 & 0 \\ 0 & \beta_{i_2} & 0 \\ 0 & 0 & \beta_{i_3} \end{bmatrix} \tag{18}$$

for $i = 0, 1, \ldots, 3$, and $\int^{(n)}$ stands for the nth integral respect to time. Here, integral error action is properly embedded into auxiliary control to actively compensate disturbances $\boldsymbol{\xi}$ as well as differences between real and reconstructed velocity vectors as described by Equation (16).

By considering the torque controllers defined by

$$\boldsymbol{\tau} = \mathbf{D}(\mathbf{q})\mathbf{v} \tag{19}$$

the closed-loop tracking error dynamics results in

$$\mathbf{e}_\mathbf{q}^{(5)} + \mathbf{B_4}\mathbf{e}_\mathbf{q}^{(4)} + \mathbf{B_3}\mathbf{e}_\mathbf{q}^{(3)} + \mathbf{B_2}\ddot{\mathbf{e}}_\mathbf{q} + \mathbf{B_1}\dot{\mathbf{e}}_\mathbf{q} + \mathbf{B_0}\mathbf{e}_\mathbf{q} = 0 \tag{20}$$

Then, the gain matrices are suitably selected as matching the following **Hurwitz** stable characteristic polynomials:

$$\mathbf{P_H}(s) = s^5\mathbf{I}_{3\times 3} + \mathbf{B}_{H_4}s^4 + \mathbf{B}_{H_3}s^3 + \mathbf{B}_{H_2}s^2 + \mathbf{B}_{H_1}s + \mathbf{B}_{H_0} \tag{21}$$

4. Artificial Neural Networks-Based Robot Motion Control

In the robust control approach introduced above, the control gains vector can be selected as a constant. However, this vector can be dynamically updated to improve the dynamic response of the system. Artificial neural networks are intelligent agents capable of learning from experiences. These artificial entities employ a massive interconnection of neurons, referred to in the literature as simple computing or processing cell units [1]. In this research, the role of the B-spline artificial neural networks is to provide adaptation capabilities to the introduced robot motion control strategy by updating online the controller gains, as shown in scheme of Figure 3. With the online training of the Bs-ANN, the control law is constantly updated based on the tracking error information. In contrast with other conventional multilayer networks, used in identification and control applications, B-spline networks are based on a defined three-layer structure which decreases its complexity, consequently making them a powerful alternative in real-time implementations.

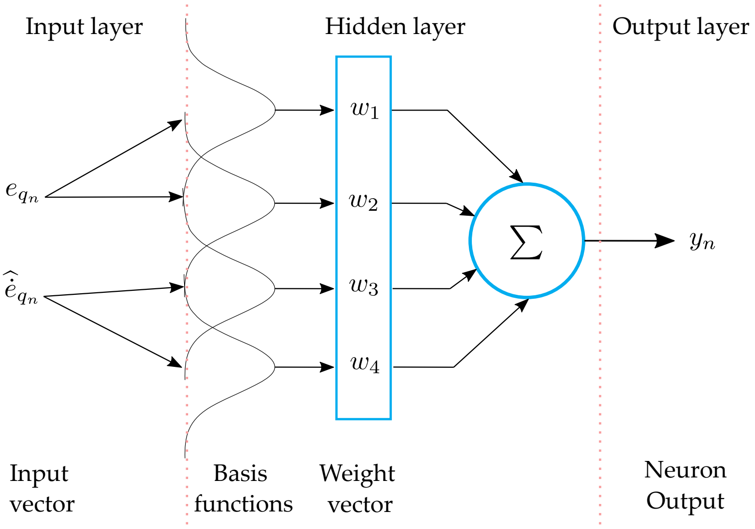


Figure 3. Proposed architecture for the B-spline artificial neural network (Bs-ANN).

Adaptive Outline for Control Purposes

Without high-gain feedback or high-frequency switching, adaptive control schemes have successfully handled parametric uncertainties and achieved asymptotic stability [32]. Using a linear combination of mono-variable and multi-variable basis functions, a B-spline function is a polynomial function defined by its extremes. In this work, an adaptive robust control scheme was dynamically tuned using B-spline artificial neural networks. Each Bs-ANN updates its synaptic weights based on different learning indexes and inputs. By continually learning the variables associated with the physical system, this type of network can cope with the system's nonlinearities and uncertainties [27].

In Figure 3, an implementation of Bs-ANN is shown as a set of associative networks with synaptic weights that can be adjusted to reproduce a specific function. The author in [33] proposes the following output

$$y_n = \mathbf{a}\mathbf{w}, \quad \mathbf{w} = \left[w_1\, w_2 \ldots w_j\right]^T, \quad \mathbf{a} = \left[a_1\, a_2 \ldots a_j\right] \tag{22}$$

where w_j and a_j are the j-th weight and the j-th basis function input, respectively, and j is the number of synaptic weights. Here, the B-spline functions are used to calculate the gains of the controllers, with the input of each neural network defined as the tracking error and the error derivatives, and with the output $y_n(t)$ defined as the control parameters. To determine the initial weight values and input thresholds, it is common to perform offline training of the network by studying the system in different and relevant operational conditions, which can be easily done by using bio-inspired optimization algorithms [26]. The minimization error, i.e., the difference between the actual output vector and its desired value, is a key element of the learning process. For this development, the following instant learning rule was adopted [34]:

$$w_n(t) = w_n(t-1) + \frac{\ell e_n(t)}{\| \mathbf{a}(t) \|_2^2}\mathbf{a}(t) \tag{23}$$

here, $\| \cdot \|_2$ is the vector 2-norm or Euclidean norm; ℓ is the learning rate, and $e_n(t)$ is the instantaneous output error. Through this method, training is performed online and continuously, with weight values updated in response to tracking error and derivative error feedback. With the boundaries properly set by choosing the correct knot vector and basis function form, this single-inner-layer structure becomes an extremely powerful mechanism. For the proposed adaptive control scheme, four third-order base functions are used: two related to the tracking error and two for the error derivative.

5. Numeric Simulation Results

The proposed motion control scheme was evaluated through several numeric simulations. During the first scenario, the proposal is implemented for motion control in joint-space mode. Secondly, the approach is used in motion control in the operative space, where the independence of a precise robot dynamical model is exhibited, and control gains are suitably updated online. Finally, the proposed control scheme is compared with a PID-like control scheme when the system is subjected to external disturbance torques. It is important to note that in the different laser manufacturing applications, the precise motion of the robot is required, even in the presence of disturbances, as addressed in the numeric experiments.

During the simulations, the anthropomorphic manipulator robot with three degrees of freedom previously described is considered, which is characterized by the set of parameters presented in Table 1.

Table 1. System parameters used during simulation scenarios.

Parameter	Quantity	Units	Description
m_1	0.73	Kg	Link 1 mass
m_2	0.85	Kg	Link 2 mass
m_3	0.51	Kg	Link 3 mass
d_1	0.14	m	Link 1 length
l_{c_2}	0.10	m	Link 2 length to the center of mass
l_2	0.20	m	Link 2 length
l_{c_3}	0.20	m	Link 3 length to the center of mass
l_3	0.30	m	Link 3 length
I_{z_1}	0.0015	Kg m^2	Inertia of link 1 in the Z axis
I_{y_2}	0.0054	Kg m^2	Inertia of link 2 in the Y axis
I_{z_2}	0.0013	Kg m^2	Inertia of link 2 in the Z axis
I_{y_3}	0.0031	Kg m^2	Inertia of link 3 in the Y axis
I_{z_3}	0.0032	Kg m^2	Inertia of link 3 in the Z axis
g	9.81	m/s ^{2}	Gravitational acceleration constant

5.1. Scenario 1: Joint Space Control

In Figure 4, it is presented the proposed control scheme for the first scenario. It is evident from this figure that only measurements of the angular positions are required for the adaptive robust controller.

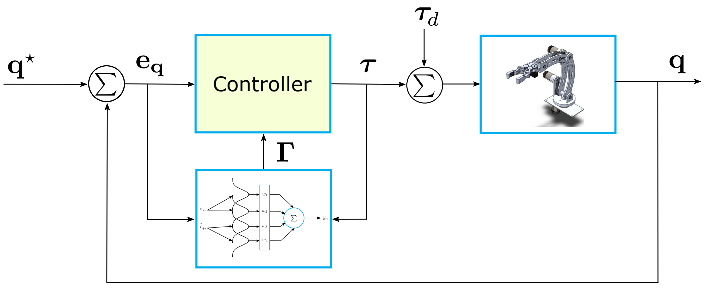


Figure 4. Motion robot control in the joint space.

On the other hand, it is relevant to mention that in order to tune a reduced number of control parameters, the following Hurwitz stable polynomial is considered for each degree of freedom:

$$P_d(s) = (s^2 + 2\zeta_c\omega_c s + \omega_c^2)^2(s + P_c) \tag{24}$$

here $\omega_c, \zeta_c, P_c > 0$, are the tuning parameters.

Therefore, to ensure closed-loop stability and proper tracking of the planned trajectory, control gains can be selected as follows:

$$\begin{aligned}
\beta_{4_i} &= 4\zeta_c\omega_c + P_c \\
\beta_{3_i} &= 2\omega_c^2 + 4\zeta_c^2\omega_c^2 + 4P_c\zeta_c\omega_c \\
\beta_{2_i} &= 4\omega_c^3\zeta_c + 2P\omega_c^2 + 4P_c\zeta_c^2\omega_c^2 \\
\beta_{1_i} &= 4P_c\omega^3\zeta_c + \omega_c^4 \\
\beta_{0_i} &= P_c\omega_c^4
\end{aligned} \tag{25}$$

As a result, only three tuning parameters are required instead of six.

During the first simulation scenario, the robot performs motion regulation in the articular space. For smooth transitions between initial and final angular positions, let us introduce the position reference profile in (26), which is implemented in order to avoid abrupt motion:

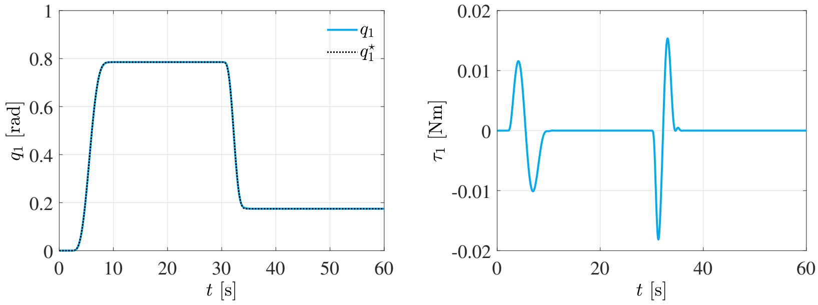
$$\Pi^\star = \begin{cases} \Pi_0 & 0 \le t < T_1 \text{ [s]} \\ \Pi_0 + \left(\Pi_f - \Pi_0\right)\mathcal{B}_z(t, T_1, T_2) & T_1 \le t \le T_2 \text{ [s]} \\ \Pi_f & t > T_2 \text{ [s]} \end{cases} \tag{26}$$

where Π_0 and Π_f stand for desired initial and final values of angular motion trajectories planned for the manipulator robot. Meantime, $\mathcal{B}_z$ is a Bézier polynomial defined as

$$\mathcal{B}_z(t, T_1, T_2) = \sum_{k=0}^{n} \delta_k \left(\frac{t - T_1}{T_2 - T_1}\right)^k \tag{27}$$

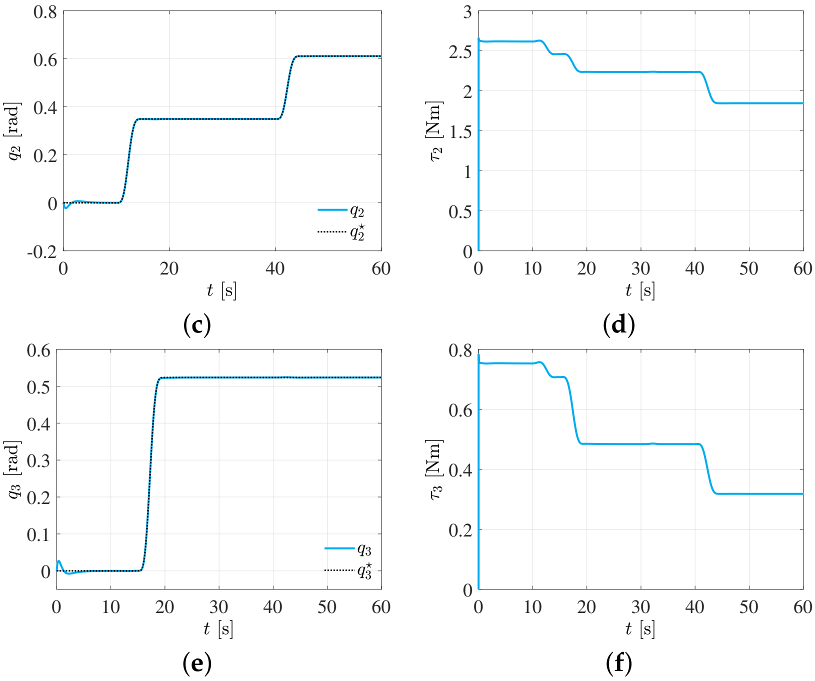
with $n = 6$, and $\delta_1 = 252$, $\delta_2 = 1050$, $\delta_3 = 1800$, $\delta_4 = 1575$, $\delta_5 = 700$, $\delta_6 = 126$ [35].

In Figure 5, it is portrayed the results for the first scenario, where several changes in the trajectory references are commanded for each joint by using the expression in (27). From the figure, it is evident a suitable tracking of the Bézier motion profiles. The computed driving torques for each joint are soft and reachable. A very common problem in the design of controllers is the saturation of the actuators due to the high magnitude of the required torque signals, something that in this case is solved satisfactorily.



(**a**) (**b**)

Figure 5. *Cont.*

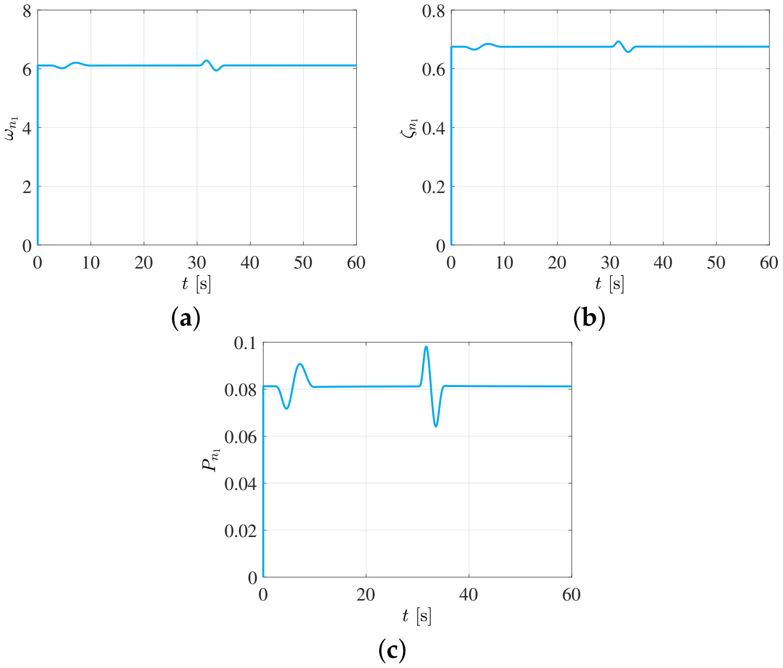


(**c**) (**d**)

(**e**) (**f**)

Figure 5. Controlled robot motion in the joint-space for the first scenario. (**a**) Controlled angular position q_1. (**b**) Computed driving torque τ_1. (**c**) Controlled angular position q_2. (**d**) Computed driving torque τ_2. (**e**) Controlled angular position q_3. (**f**) Computed driving torque τ_3.

Notwithstanding, the gain values are based on the computation of the ω_c, ζ_c, P_c parameters, as shown in Figures 6 and 7. In this case of joint movement control, the evolution of the control parameters in the first joint is presented. This is because it is the joint that bears the full weight of the robot, so it would be expected to require a greater magnitude of torque to perform the required movements. However, due to the proposed control scheme, it should be noted that the demanded torque has a very small magnitude, and an excellent follow-up of the programmed joint trajectory is obtained.



(**a**) (**b**)

(**c**)

Figure 6. Computed dynamic control parameters for the first joint motion (**a**) Adaptive ω_{n_1} control parameter. (**b**) Adaptive ζ_{n_1} control parameter. (**c**) Adaptive P_{n_1} control parameter.

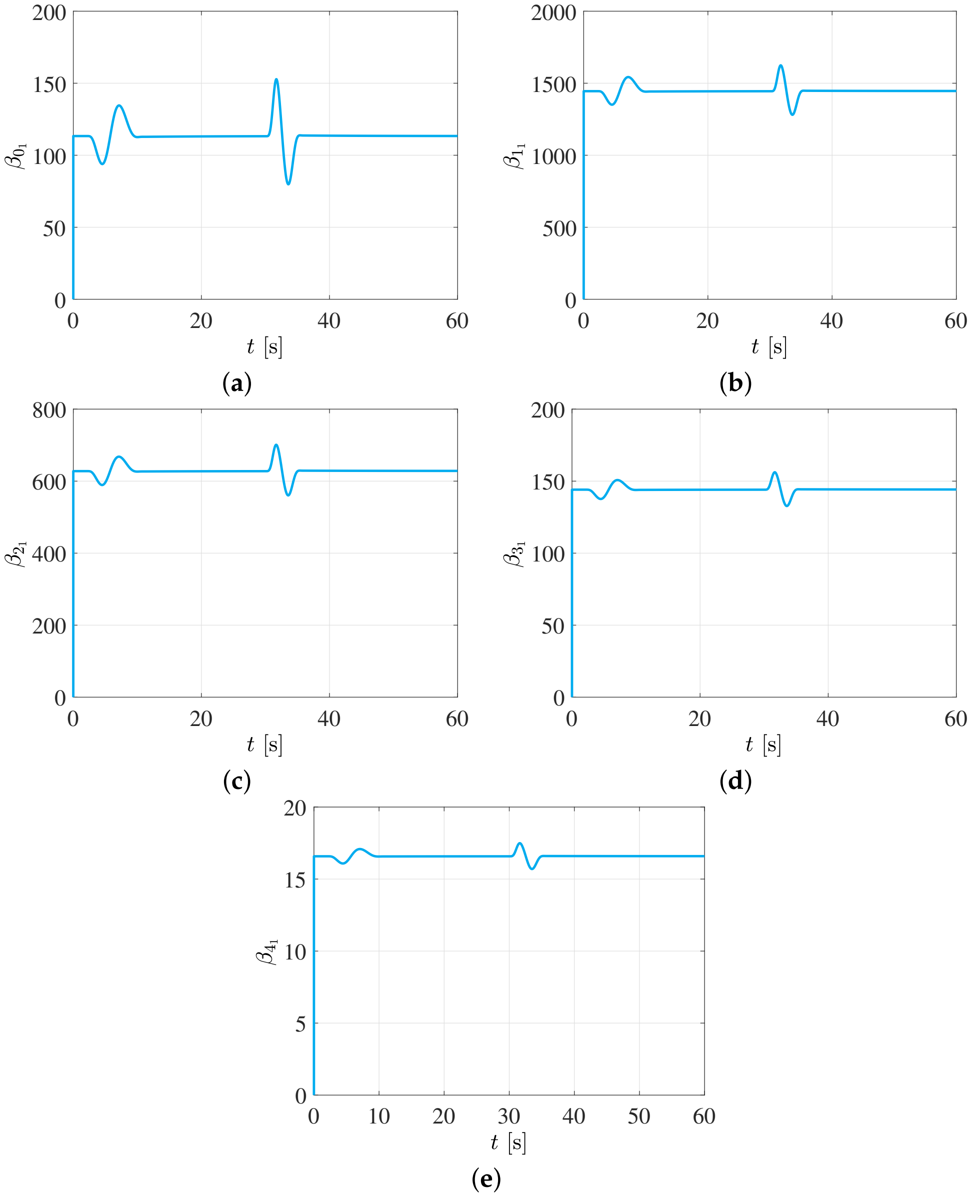


Figure 7. Computed control gains for the first joint motion. (**a**) β_{0_1}. (**b**) β_{1_1}. (**c**) β_{2_1}. (**d**) β_{3_1}. (**e**) β_{4_1}.

Notice from figures that the control gains are dynamic, updated online by the adaptive scheme based on the information of the tracking error dynamics, as shown in Figure 3.

5.2. Scenario 2: Cartesian Space Control

In the second scenario, the Cartesian space control is considered. Here, the inverse kinematics of the robot is used for computing the joint tracking references. In this experiment, a Cartesian reference trajectory is selected that is typically used in laser applications. The Cartesian position references $\mathbf{\Lambda}^{\star}$ are as follows:

$$\begin{aligned} x^{\star} &= 0.08 + 0.15(t_k/4)\ [\mathrm{m}] \\ y^{\star} &= -0.15 + 0.05\cos(\pi + t_k)\sin(\pi + t_k)\ [\mathrm{m}] \\ z^{\star} &= 0.4 - 0.1\cos(\pi + t_k)\cos(\pi + t_k)\ [\mathrm{m}] \end{aligned} \tag{28}$$

where $t_k = (t_k + \Delta_{tk})/10$ s, with initial values of t_k and Δ_{tk} are zero and updated each iteration as $\Delta_{tk} = \Delta_{tk} + 0.1 \times 10^{-3}$.

In the same way as in the previous case, it is only necessary to know the desired angular positions of the robot in order to carry out the trajectory tracking satisfactorily. Figure 8 presents the block diagram of the proposed control scheme in the Cartesian space.

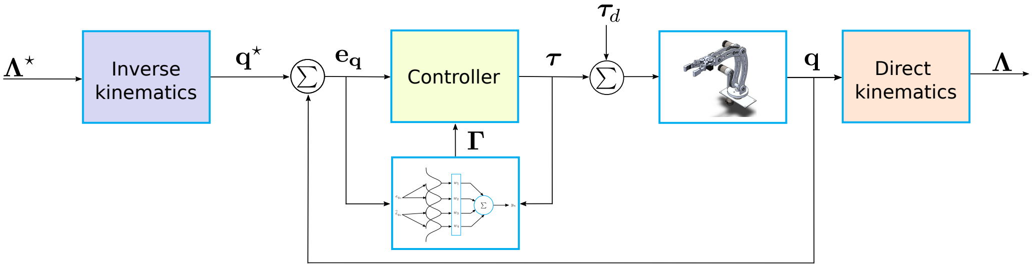


Figure 8. Motion robot control in the Cartesian space.

Figure 9 presents the results of the robot motion control scheme in Cartesian space. It is highlighted that the proposed scheme presents excellent tracking results without the need to provide the speeds and accelerations of the reference to be followed.

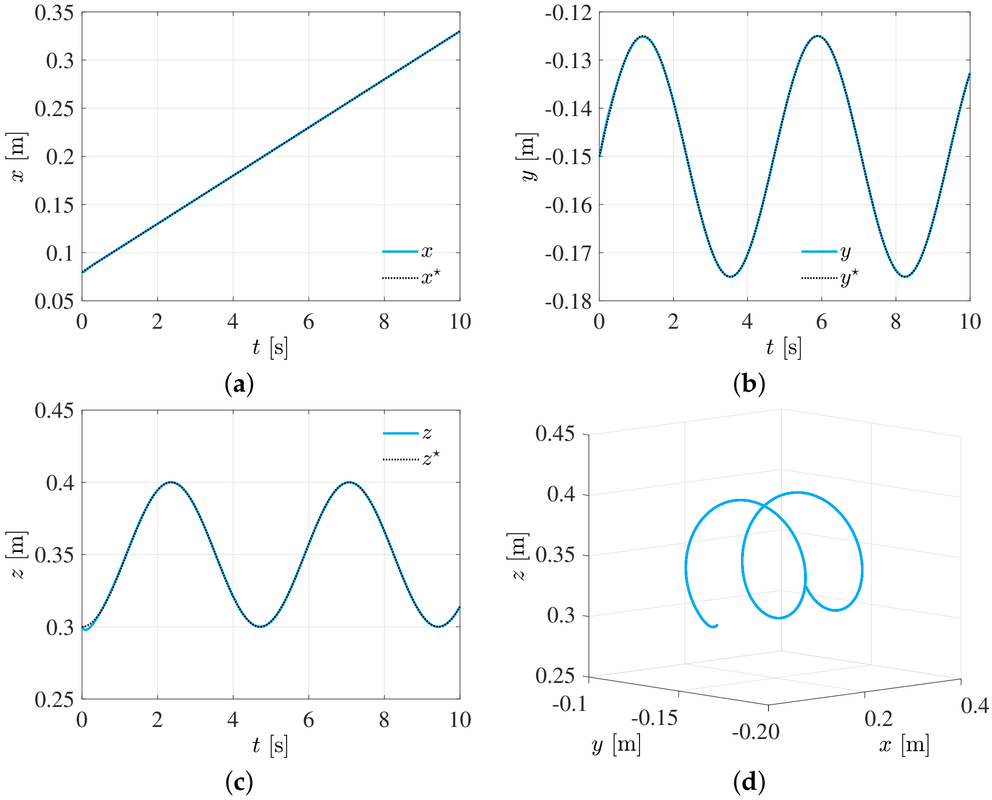


Figure 9. Motion robot control in the Cartesian space. (**a**) Trajectory tracking on the x axis. (**b**) Trajectory tracking on the y axis. (**c**) Trajectory tracking on the z axis. (**d**) 3D trajectory tracking.

Finally, Figure 10 presents the tracking of the system in the joint space. It can be noted that the system demands very small torque magnitudes to execute the movement of the system. This is a very important aspect to prevent the actuators from saturation due to very high magnitudes of torque.

In scenarios where non-modeled dynamics and/or disturbances occur, the proposed control scheme can present excellent results, so it would be an excellent alternative to be implemented in high-precision manufacturing systems, such as laser cutting and welding robots. This aspect can be better appreciated in the following scenario.

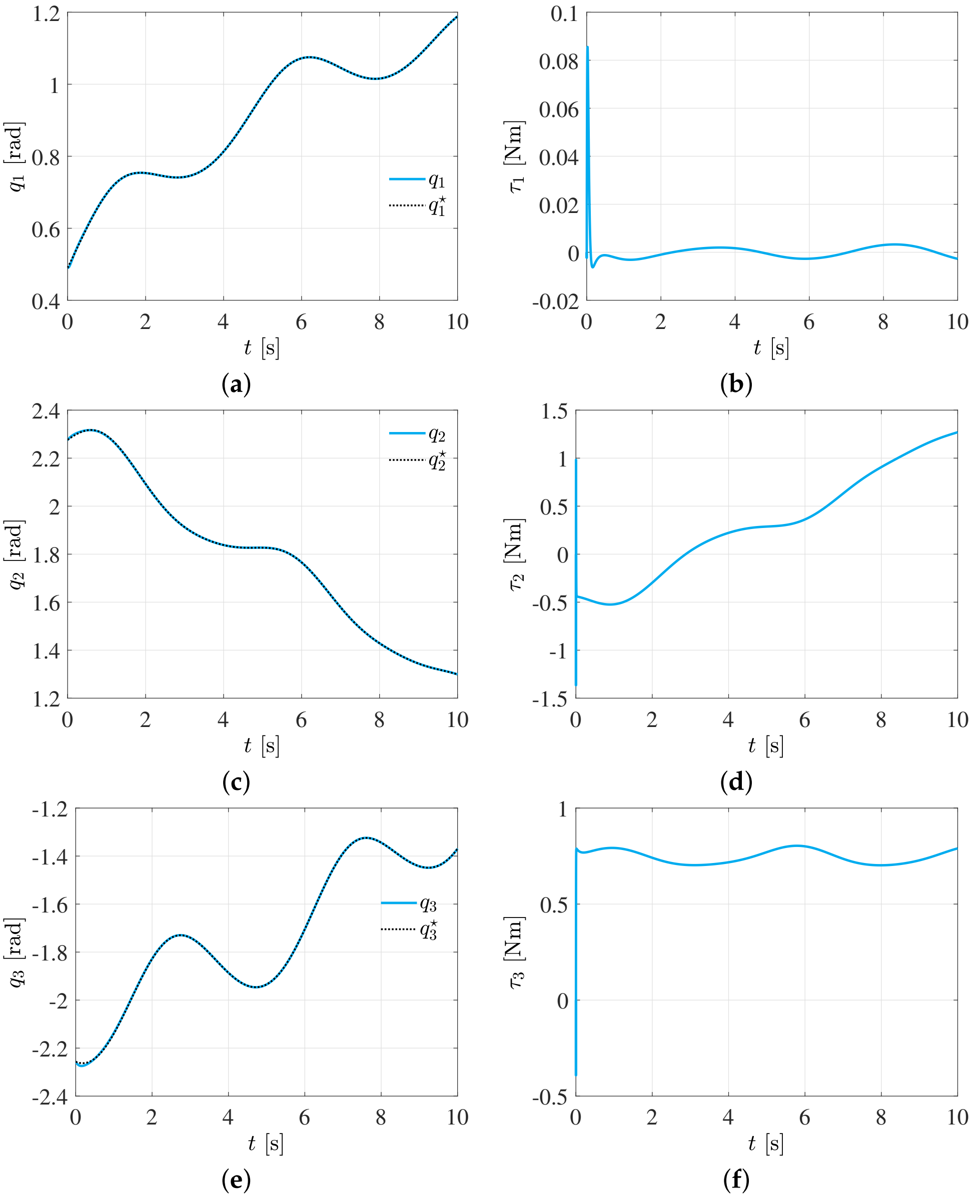


Figure 10. Properly controlled motion even in presence of unmodeled dynamics. (**a**) Controlled q_1 joint motion. (**b**) Computed driving torque τ_1. (**c**) Controlled q_2 joint motion. (**d**) Computed driving torque τ_2. (**e**) Controlled q_3 joint motion. (**f**) Computed driving torque τ_3.

5.3. Scenario 3: Joint Space Robot Motion Subjected to External Vibrating Torques

To demonstrate the effectiveness of the proposed control strategy, the joint space motion control of the manipulator robot subjected to disturbance torques $\boldsymbol{\tau}_d = \left[\tau_{d_1} \tau_{d_2}\ \tau_{d_3}\right]^{\mathsf{T}}$ is presented. Here, the robotic system is intentionally suddenly perturbed with unknown and undesired vibrating torques, portrayed in Figure 11, which are given as follows:

$$\tau_{d_1} = \begin{cases} 0\,[\text{Nm}] & 0 \le t < 10\,[\text{s}] \\ 0.7\cos(0.1t)\,[\text{Nm}] & t \ge 10\,[\text{s}] \end{cases} \tag{29}$$

$$\tau_{d_2} = \begin{cases} 0\,[\text{Nm}] & 0 \le t < 15\,[\text{s}] \\ 0.3\cos(0.1t) + 0.2\sin(0.5t)\,[\text{Nm}] & t \ge 15\,[\text{s}] \end{cases} \tag{30}$$

$$\tau_{d_3} = \begin{cases} 0\,[\text{Nm}] & 0 \le t < 20\,[\text{s}] \\ 0.2\sin(0.1t) + 0.1\sin(0.5t)\,[\text{Nm}] & t \ge 20\,[\text{s}] \end{cases} \tag{31}$$

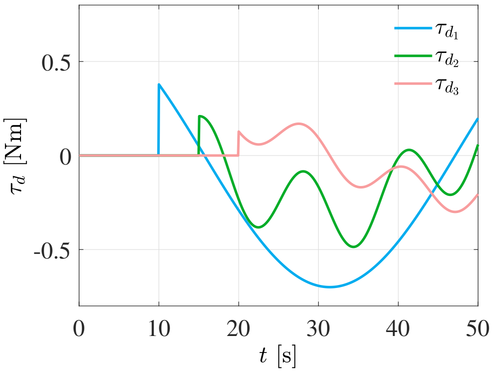


Figure 11. Induced external vibrating torques.

On the other hand, the model-based PID-like plus a dynamic inversion controller introduced in Equation (32) is presented for contrasting the capabilities of the introduced motion controller.

$$\boldsymbol{\tau} = \mathbf{D}(\mathbf{q})[\ddot{\mathbf{q}}^{\star} - \mathbf{u}] + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \tag{32}$$

where $\mathbf{u}$ is a PID-like controller vector given by

$$\mathbf{u} = \mathbf{K}_p\mathbf{e}_\mathbf{q} + \mathbf{K}_i \int_{t_0}^{t} \mathbf{e}_\mathbf{q}\, dt + \mathbf{K}_d\dot{\mathbf{e}}_\mathbf{q} \tag{33}$$

and $\mathbf{e}_\mathbf{q} = \mathbf{q} - \mathbf{q}^{\star}$. Notice from the equations that there exists a high dependence of the mathematical model, which is not desirable in manufacturing applications due to the parameter uncertainty of both the robot and the objects it must manipulate and the presence of the non-modeled dynamics and unknown perturbations, which is the case of frictions and vibrations that can occur during the operation of the system.

In Figure 12, the unperturbed responses are portrayed for both controllers. It is evident that an acceptable performance is achieved. Notwithstanding, in the presence of undesired and unknown vibration torques, the PID-like controller is unable to stabilize the system or to track the reference, as observed in Figure 13. Here, it is observed that the motion controller performance is significantly deteriorated when external disturbances are present. Notice that in the proposed motion control scheme, the suitable integration of the Bézier polynomials as the reference motion profiles allows to achieve a superior robot motion performance.

Figure 12 shows the comparison in the robot motion control between a PID-like controller dependent on the mathematical model of the system and the proposed robust adaptive controller without considering external disturbances. In the trajectory tracking with the PID-like controller, it is clearly observed that the system does not reach the desired reference with the requirements that are being demanded, but instead presents a delay, which is not desirable in high-precision manufacturing applications. In contrast, the proposed control strategy reaches the references with smooth and controlled movements, which allows to avoid collisions and damages in cases of application in real environments both for the robot and also for what it is going to manipulate. A system with a PID-like controller in most cases can only work under pre-established conditions.

It is also true that the tuning of the PID controller gains could be improved, but this does not make it less dependent on the model and due to the non-modeled dynamics existing in the robot environment, it would still not work properly if subjected to external disturbances. In addition, a greater number of control parameters than those calculated with the proposed controller would have to be considered.

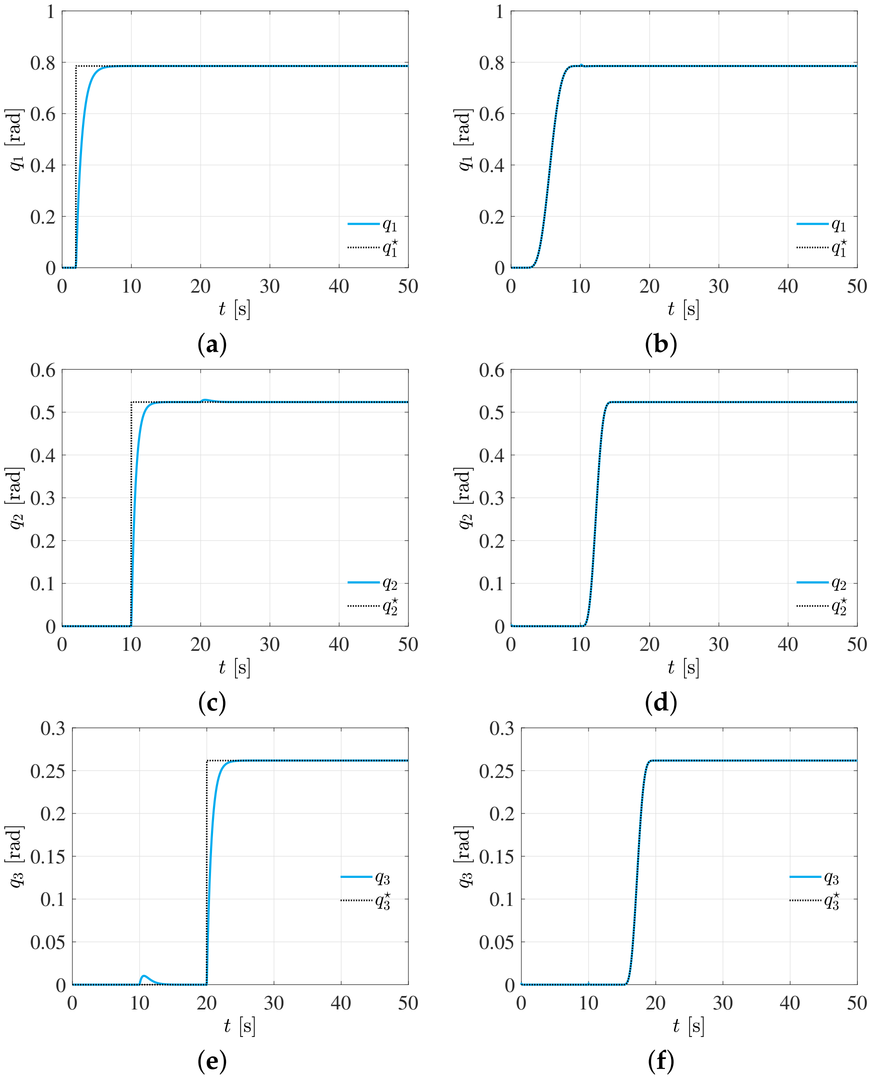


Figure 12. Controlled motion without external disturbance inputs. (**a**) Controlled q_1 joint motion with PID controller. (**b**) Controlled q_1 joint motion with proposed robust adaptive controller. (**c**) Controlled q_2 joint motion with PID controller. (**d**) Controlled q_2 joint motion with proposed robust adaptive controller. (**e**) Controlled q_3 joint motion with PID controller. (**f**) Controlled q_3 joint motion with proposed robust adaptive controller.

Figure 13 shows that when the system with the PID controller is subjected to external disturbances, it has a very low performance, so the robot does not have the ability to perform the required trajectory tracking, unlike the proposed control scheme in which it can be seen that no deviation is generated from the paths that the robot is required to follow despite the disturbances that are being input.

Finally, it is considered a last experiment, where parametric uncertainty in matrix $\mathbf{D}(\mathbf{q})$ is presented, which might affect considerably the system performance since the controllers introduced in Equations (19) and (32) depend on the matrix values. In this fashion, let us consider additional variations of ±20% in the matrix value used for control purposes under the previous scenario conditions. A quantitative comparison of the controllers performance is carried out by using the integral time absolute error (ITAE) and the integral squared control input (ISCI) indexes, which are associated with the tracking error and control input efforts for each joint as follows:

$$\mathrm{ITAE} = \int_0^t t|e|dt \qquad (34)$$

$$\text{ISCI} = \int_0^t u^2 dt \tag{35}$$

where e and u stand for the tracking errors and control inputs for each degree of freedom. From Table 2, it is corroborated a better performance of the proposed control scheme for the robotic manipulator systems, even when less information is required from the physical system in contrast with the PID-like controller. Moreover, in Tables 3 and 4, it is summarized the performance indexes when there exist variations in the matrix values used by the control schemes, which are ±20 of the nominal real matrix value. Furthermore, from Equations (19) and (32), it is evident that when uncertainty is also considered in the Coriolis and centripetal forces matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, the performance of the PID-like controller is further deteriorated. On the other hand, the low system model information dependence of the proposed adaptive robust motion tracking controller allows the system to portray a better performance when the system is subjected to parametric uncertainty, non-modeled dynamics and unknown disturbing vibrating load torques. Overall, the presented results demonstrate that the proposal is a feasible and robust alternative to achieve an acceptable performance in motion tracking control for robotic manipulator systems.

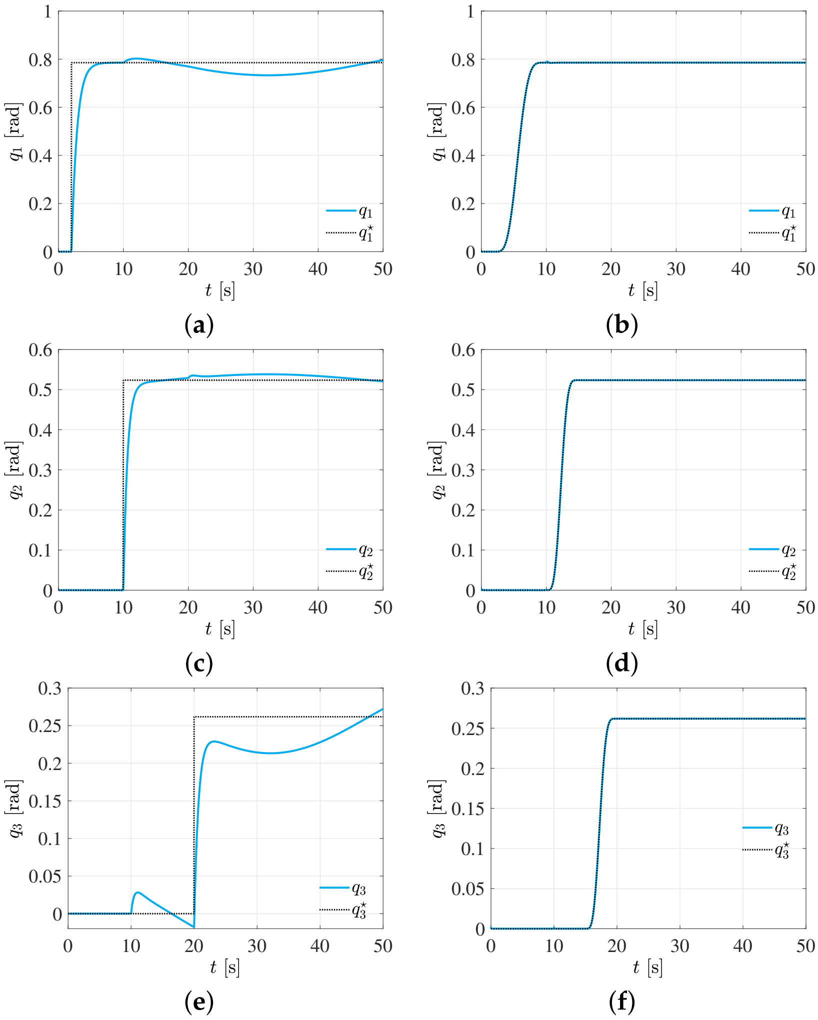


Figure 13. Controlled motion subjected to external disturbance inputs. (**a**) Controlled q_1 joint motion with PID controller. (**b**) Controlled q_1 joint motion with proposed robust adaptive controller. (**c**) Controlled q_2 joint motion with PID controller. (**d**) Controlled q_2 joint motion with proposed robust adaptive controller. (**e**) Controlled q_3 joint motion with PID controller. (**f**) Controlled q_3 joint motion with proposed robust adaptive controller.

Table 2. 0% variation from the real $\mathbf{D}(\mathbf{q})$ matrix in the controller implementation.

	PID-Like		Proposal	
Joint	**ITAE**	**ISCI**	**ITAE**	**ISCI**
q_1	37.08	8.75	0.05	8.02
q_2	13.50	323.68	2.60×10^{-3}	322.76
q_3	36.54	41.11	3.80×10^{-4}	39.51

Table 3. 20% variation from the real $\mathbf{D}(\mathbf{q})$ matrix in the controller implementation.

	PID-Like		Proposal	
Joint	**ITAE**	**ISCI**	**ITAE**	**ISCI**
q_1	45.85	8.64	0.04	8.02
q_2	16.10	324.08	2.21×10^{-3}	322.67
q_3	45.06	41.33	2.81×10^{-4}	39.77

Table 4. −20% variation from the real $\mathbf{D}(\mathbf{q})$ matrix in the controller implementation.

	PID-Like		Proposal	
Joint	**ITAE**	**ISCI**	**ITAE**	**ISCI**
q_1	31.22	8.84	3.01	35.53
q_2	11.78	324.10	0.08	324.4972
q_3	30.91	41.06	0.0014	39.77

6. Conclusions

In this paper, the anthropomorphic robot manipulation problem was addressed with an adaptive robust motion control scheme, where B-spline artificial neural networks and dynamic compensators are successfully integrated in motion controllers for regulation and output tracking tasks, where just the angular position information is demanded. In this contribution, it is neither required the use of a disturbance observer, angular velocity measurements nor the feed-forward of the tracking references information. The performance of the proposed controller is demonstrated in exhaustive simulation experiments where the system is subjected to both external undesired forced torques and non-modeled dynamics. Acceptable levels of the tracking error are corroborated when the proposed motion control scheme is implemented in a three degree of freedom virtual robot system in the articular and Cartesian space perspectives. It is worth mentioning that this novel proposal can be easily further extended to n-degrees of freedom robotic manipulator systems. Future research work will deal with the synthesis of high-efficiency adaptive output-feedback planned motion profile tracking control strategies based on artificial neural networks for MIMO nonlinear robotic systems under operational scenarios where uncertainty on dimensional specifications of mechanical design could be expected.

Author Contributions: Conceptualization, D.G.-P., H.Y.-B., F.B.-C., A.F.-C. and R.T.-O.; Data curation, D.G.-P. and H.Y.-B.; Formal analysis, F.B.-C., A.F.-C. and R.T.-O.; Investigation, D.G.-P., H.Y.-B., F.B.-C., I.R.-C., A.F.-C. and R.T.-O.; Methodology, D.G.-P., H.Y.-B., F.B.-C., A.F.-C. and R.T.-O.; Project administration, F.B.-C.; Resources, I.R.-C.; Software, D.G.-P. and H.Y.-B.; Supervision, F.B.-C. and R.T.-O.; Validation, D.G.-P. and H.Y.-B.; Writing—original draft, D.G.-P. and H.Y.-B.; Writing—review and editing, H.Y.-B. and F.B.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank Consejo Nacional de Ciencia y Tecnología (CONACyT) and Universidad Politécnica de Tulancingo for their support given to developing this work.

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

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial Neural Networks
Bs-ANN	B-spline Artificial Neural Networks
DOF	Degrees of Freedom
PID	Proportional Integral Derivative

References

1. Haykin, S. *Neural Networks and Learning Machines, 3/E*; Pearson Education, Inc.: Upper Saddle River, NJ, USA, 2009.
2. Lewis, F.; Jagannathan, S.; Yesildirak, A. *Neural Network Control of Robot Manipulators and Non-Linear Systems*; CRC Press: Boca Raton, FL, USA, 2020.
3. Ibrahim, K.; Sharkawy, A.B. A hybrid PID control scheme for flexible joint manipulators and a comparison with sliding mode control. *Ain Shams Eng. J.* **2018**, *9*, 3451–3457. [CrossRef]
4. Beschi, M.; Mutti, S.; Nicola, G.; Faroni, M.; Magnoni, P.; Villagrossi, E.; Pedrocchi, N. Optimal robot motion planning of redundant robots in machining and additive manufacturing applications. *Electronics* **2019**, *8*, 1437. [CrossRef]
5. Jin, L.; Li, S.; Luo, X.; Li, Y.; Qin, B. Neural dynamics for cooperative control of redundant robot manipulators. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3812–3821. [CrossRef]
6. Zhu, Y.; He, X.; Liu, Q.; Guo, W. Semiclosed-loop motion control with robust weld bead tracking for a spiral seam weld beads grinding robot. *Robot. Comput.-Integr. Manuf.* **2022**, *73*, 102254. [CrossRef]
7. Wang, F.; Chao, Z.q.; Huang, L.b.; Li, H.y.; Zhang, C.q. Trajectory tracking control of robot manipulator based on RBF neural network and fuzzy sliding mode. *Clust. Comput.* **2019**, *22*, 5799–5809. [CrossRef]
8. Yañez Badillo, H.; Tapia Olvera, R.; Aguilar Mejía, O.; Beltrán Carbajal, F. Control neuronal en línea para regulación y seguimiento de trayectorias de posición para un quadrotor. *Rev. Iberoam. Automática E Inform. Ind.* **2017**, *14*, 141–151. [CrossRef]
9. Aguilar-Mejía, O.; Tapia-Olvera, R.; Valderrabano-González, A.; Rivas Cambero, I. Adaptive neural network control of chaos in permanent magnet synchronous motor. *Intell. Autom. Soft Comput.* **2016**, *22*, 499–507. [CrossRef]
10. Dongare, A.D.; Kharde, R.R.; Kachare, A.D. Introduction to artificial neural network. *Int. J. Eng. Innov. Technol. (IJEIT)* **2012**, *2*, 189–194.
11. Bohra, P.; Campos, J.; Gupta, H.; Aziznejad, S.; Unser, M. Learning activation functions in deep (spline) neural networks. *IEEE Open J. Signal Process.* **2020**, *1*, 295–309. [CrossRef]
12. 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]
13. Cao, S.; Sun, L.; Jiang, J.; Zuo, Z. Reinforcement Learning-Based Fixed-Time Trajectory Tracking Control for Uncertain Robotic Manipulators with Input Saturation. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–12. [CrossRef] [PubMed]
14. Chen, S.; Wen, J.T. Industrial Robot Trajectory Tracking Control Using Multi-Layer Neural Networks Trained by Iterative Learning Control. *Robotics* **2021**, *10*, 50. [CrossRef]
15. Elsisi, M.; Mahmoud, K.; Lehtonen, M.; Darwish, M.M. An improved neural network algorithm to efficiently track various trajectories of robot manipulator arms. *IEEE Access* **2021**, *9*, 11911–11920. [CrossRef]
16. Yang, C.; Huang, D.; He, W.; Cheng, L. Neural control of robot manipulators with trajectory tracking constraints and input saturation. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4231–4242. [CrossRef] [PubMed]
17. Zhou, M.; Feng, Y.; Xue, C.; Han, F. Deep convolutional neural network based fractional-order terminal sliding-mode control for robotic manipulators. *Neurocomputing* **2020**, *416*, 143–151. [CrossRef]
18. Nohooji, H.R. Constrained neural adaptive PID control for robot manipulators. *J. Frankl. Inst.* **2020**, *357*, 3907–3923. [CrossRef]
19. Dachang, Z.; Baolin, D.; Puchen, Z.; Shouyan, C. Constant force PID control for robotic manipulator based on fuzzy neural network algorithm. *Complexity* **2020**, *2020*, 3491845. [CrossRef]
20. Shojaei, K.; Kazemy, A.; Chatraei, A. An Observer-Based Neural Adaptive PID Controller for Robot Manipulators Including Motor Dynamics With a Prescribed Performance. *IEEE/ASME Trans. Mechatron.* **2020**, *26*, 1689–1699. [CrossRef]
21. Yañez-Badillo, H.; Beltran-Carbajal, F.; Tapia-Olvera, R.; Valderrabano-Gonzalez, A.; Favela-Contreras, A.; Rosas-Caro, J.C. A Dynamic Motion Tracking Control Approach for a Quadrotor Aerial Mechanical System. *Shock Vib.* **2020**, *2020*, 6635011. [CrossRef]
22. Beltran-Carbajal, F.; Tapia-Olvera, R. An adaptive neural online estimation approach of harmonic components. *Electr. Power Syst. Res.* **2020**, *186*, 106406. [CrossRef]
23. Tapia, O.; Ramirez, J.M. Power Systems Neural Voltage Control by a Statcom. In Proceedings of the 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, Canada, 16–21 July 2006; pp. 2249–2254.
24. Beltran-Carbajal, F.; Tapia-Olvera, R.; Lopez-Garcia, I.; Guillen, D. Adaptive dynamical tracking control under uncertainty of shunt DC motors. *Electr. Power Syst. Res.* **2018**, *164*, 70–78. [CrossRef]

25. Beltran-Carbajal, F.; Tapia-Olvera, R.; Valderrabano-Gonzalez, A.; Lopez-Garcia, I. Adaptive neuronal induction motor control with an 84-pulse voltage source converter. *Asian J. Control* **2021**, *23*, 1603–1616. [CrossRef]
26. Yañez-Badillo, H.; Beltran-Carbajal, F.; Tapia-Olvera, R.; Favela-Contreras, A.; Sotelo, C.; Sotelo, D. Adaptive Robust Motion Control of Quadrotor Systems Using Artificial Neural Networks and Particle Swarm Optimization. *Mathematics* **2021**, *9*, 2367. [CrossRef]
27. Yañez-Badillo, H.; Tapia-Olvera, R.; Beltran-Carbajal, F. Adaptive Neural Motion Control of a Quadrotor UAV. *Vehicles* **2020**, *2*, 468–490. [CrossRef]
28. Spong, M.W.; Hutchinson, S.; Vidyasagar, M. *Robot Modeling and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2020.
29. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. Kinematics. *Robotics: Modelling, Planning and Control*; Springer: New York, NY, USA, 2009; pp. 39–103.
30. Cortés, F.R. *Robótica: Control de Robots Manipuladores*; Marcombo: Barcelona, Spain, 2020.
31. Fliess, M.; Marquez, R.; Delaleau, E.; Sira-Ramirez, H. Correcteurs proportionnels-intégraux généralisés. *ESAIM Control Optim. Calc. Var.* **2002**, *7*, 23–41. [CrossRef]
32. Yao, J.; Deng, W. Active disturbance rejection adaptive control of uncertain nonlinear systems: Theory and application. *Nonlinear Dyn.* **2017**, *89*, 1611–1624. [CrossRef]
33. Bossley, K.; Brown, M.; Harris, C. Neurofuzzy adaptive modelling and construction of nonlinear dynamical processes. In *Neural Network Applications in Control*; The Institution of Electrical Engineers: London, UK, 1995; pp. 253–292.
34. Saad, D. *On-Line Learning in Neural Networks*; Cambridge University Press: Cambridge, UK, 1999.
35. Beltran-Carbajal, F.; Silva-Navarro, G. Output feedback dynamic control for trajectory tracking and vibration suppression. *Appl. Math. Model.* **2020**, *79*, 793–808. [CrossRef]