Journal of Intelligent & Robotic Systems
https://doi.org/10.1007/s10846-020-01242-9

A New Adaptive RISE Feedforward Approach based on Associative Memory Neural Networks for the Control of PKMs

Jonatan Martín Escorcia-Hernández[1] · Hipólito Aguilar-Sierra[2] · Omar Aguilar-Mejia[3] · Ahmed Chemori[4] · José Humberto Arroyo-Núñez[1]

Received: 9 December 2019 / Accepted: 29 July 2020
© Springer Nature B.V. 2020

Abstract

In this paper, a RISE (Robust Integral of the Sign Error) controller with adaptive feedforward compensation terms based on Associative Memory Neural Network (AMNN) type B-Spline is proposed to regulate the positioning of a Delta Parallel Robot (DPR) with three degrees of freedom. Parallel Kinematic Manipulators (PKMs) are highly nonlinear systems, so the design of a suitable control scheme represents a significant challenge given that these kinds of systems are continually dealing with parametric and non-parametric uncertainties and external disturbances. The main contribution of this work is the design of an adaptive feedforward compensation term using B-Spline Neural Networks (BSNNs). They make an on-line approximation of the DPR dynamics and integrates it into the control loop. The BSNNs' functions are bounded according to the extreme values of the desired joint space trajectories that are the BSNNs' inputs, and their weights are on-line adjusted by gradient descend rules. In order to evaluate the effectiveness of the proposed control scheme with respect to the standard RISE controller, numerical simulations for different case studies under different scenarios were performed.

Keywords Delta parallel robot · RISE control · B-spline neural network · Trajectory tracking · On-line learning

1 Introduction

PKMs have gained significant interest in recent decades thanks to their desired features provided by their construction based on several closed-loop kinematic chains [1]. This configuration provides some advantages to PKMs over their serial counterparts. For instance, the overall stiffness in PKMs is higher than concerning serial manipulators owing to several limbs joined to a fixed base to support the traveling plate where the end-effector is located, generating more resistance against the deflections caused by external forces or moments exerted on the end-effector [2]. Besides, this arrangement allows to PKMs to obtain absolute greater accuracy, better repeatability, more capacity to carry heavier loads, and the ability to execute faster and more precise movements [3]. These features make PKMs attractive solutions for tasks that require high positioning accuracy and precision, and for these reasons are widely used in product transportation and classification tasks, haptic devices, agricultural applications, machine tools, laser cutting, 3D printers, among others [4], [5], [6]. One of the most studied PKM in the literature is the DPR developed in the 80's by Reymond Clavel. [7]. The main distinction of the DPR other existing PKMs concepts is the use of mechanisms based on parallelograms. The parallelograms restrain the orientation of the traveling plate entirely resulting in only translational movements over the three axes of the Cartesian space. Besides, its closed kinematic chains are very light, allowing this robot to reach high extreme accelerations. For these features, the DPR is mainly used in Pick and Place (P&P) tasks [4]. However, the operational workspace of PKMs is reduced in comparison to Serial Manipulators. Moreover, PKMs are known for their highly nonlinear dynamics, which is increases considerably when the PKM is operated at high speeds/accelerations leading to mechanical vibration issues [8]. Additionally, the closed-loop configuration yields coupling dynamics; therefore, the actuators must work in complete synchronization with each other for not damaging the PKMs' mechanism. The previous problem is closely related to unstructured or/and structured uncertainties. Geometric errors, sensors noise, components degradation, and modeling simplifications, e.g., not considered friction or actuator dynamics, are considered the first

✉ Hipólito Aguilar-Sierra
hipolito.aguilar@lasalle.mx

Extended author information available on the last page of the article.

kind of uncertainties. The second kind of uncertainties is generated by parameter variations owing to operate environment or inaccurate knowledge of dynamic parameters [9]. For the PKMs to perform tasks satisfactorily, advanced control techniques should be considered to overcome the issues and challenges mentioned above, guaranteeing the minimum possible tracking error [10]. To deal with the discussed control challenges for PKMs, we propose a RISE controller with an adaptive feedforward term based on AMNNs. The main contribution of the paper is the design of an adaptive feedforward compensation term based on BSNNs. They make an on-line approximation of the DPR dynamics and integrated it into the control-loop. The BSNNs' functions are bounded according to the extreme values o the desired joint space trajectories that are the BSNNs' inputs, and their weights are on-line adjusted by gradient descend rules. The remainder of this paper is organized as follows: In Section 2, the state of the art of proposed control solutions for robotics emphasizing in PKMs is presented. In Section 3 the kinematic and dynamic models of a DPR are presented. In Section 4, the proposed RISE controller with adaptive BSNN compensation is set out in detail. To know the effectiveness of the proposed control scheme, numerical simulations are presented in Section 5, where the control system is proven in two case studies under various scenarios. Finally, conclusions are detailed in Section 6.

2 State of the Art

For PKMs, several control techniques have been developed and implemented to deal with the previously mentioned challenges, highlighting conventional feedback controllers, nonlinear controllers, robust controllers, adaptive controllers, or a combination of them [2]. Control schemes based on the PD/PID feedback control have been extensively used for control of PKMs, due to its easy implementation and its relatively good performance. However, in PKMs, the performance of this type of controllers decreases notoriously when the system is subjected to sudden changes in the acceleration and dynamic parameter variation [11], [12]. Robust linear control techniques such as the H^{∞} are used for systems affected by the presence of external disturbances and parametric variations [13]. An efficient implementation of a H^{∞} multivariable controller PKMs is presented in [14]; in such scheme, a linearized model around an operating equilibrium point is determined to obtain a state-space representation of the DPR, besides that, the sensitivity and complementary sensitivity transfer functions are calculated. This technique utilizes the perturbations in the design of the controller, but the design of this scheme is very sophisticated and complex. RISE is a novel robust nonlinear feedback control technique that is becoming popular in robotics control. This control scheme outcomes limitations presented in PD/PID controllers thanks to its robust nonlinear term, and it ensures semi-global asymptotic stability in the presence of general uncertain disturbances [15] besides, its implementation is straightforward without many complications as other robust techniques. This control law has been implemented satisfactorily in PKMs, as was demonstrated in [16]. Some modifications have been made to the original RISE control to improve its qualities, e.g., in [17], a RISE control with nonlinear gains was proposed to regulate the position of a DPR. Moreover, RISE control is suitable to be combined with model-based terms to enhance the overall system performance, as was demonstrated in [18], where a RISE controller with computed feedforward was proposed to regulate the trajectory tracking of a PKM designed for machining operations. However, for model-based controllers, the lack of accurate knowledge of parameters may lead to degrading the controller efficiency instead of improving it. Adaptive controllers have been proposed to deal with the above problems. These control schemes started from the issue that some dynamic model elements are not accurately known. They included an adaptation rule which adjusts controller parameters to changes in the controlled system according to given criteria [19]. In [20], a RISE controller with adaptive feedforward was proposed to control a redundantly actuated PKM dealing with the issue of parametric uncertainties. We can mention other adaptive control proposals solutions making use of artificial intelligence. For instance, in [21], a reinforcement learning with a complete inverse kinematic solution was proposed to balance the lower body of an NAO robot. This control solution can compensate external disturbances modifying its value function parameters. In [22], a model-free adaptive controller was proposed to control a pneumatic actuator. The controller makes use of a Q-function to estimate the long-term performance of the adaptive control. This solution can stabilize the system in the presence of nonparametric and parametric uncertainties. Some adaptive controllers make use of Artificial Neural Networks (ANNs) to approximate unknown nonlinear dynamics and integrated it into the control-loop [23]. In the literature, it has been reported several adaptive control schemes based on ANNs applied to robotics control. We can distinguish two architectures of ANN. The first one is the multi-layer ANN. This configuration increases the computation complexity since the information travels bidirectionally between the hidden layers of the neural network, besides they entail a considerable computational cost requiring long training time [24]. The second one is the single-layer ANN. This kind of ANN requires less computational process due to its single layer of neurons; the AMNN belongs to this configuration. These kinds of ANN assume the principle of local generalization, implying that for a specific input, just a portion of the ANN will be involved; thus, the computational effort is reduced. Moreover, their activation functions are linear respect to the adaptable weights so, straightforward instantaneous learning rules can be used to update their adjusted weights [25]. There have been some recent advances in the field of robotics control using ANN. In the branch of multi-layer-based ANN, a nonlinear adaptive controller was proposed to regulate the trajectory tracking of a Cable-driven robot in [26]; the controller can compensate for parametric and non-parametric uncertainties of the nonlinear robot dynamics; the weights are updated trough projection operators. Besides, it has been reported several control schemes based on single-layer ANNs. In [27], a modified version Cerebellar Model Articulation Controller (CMAC) was proposed to find optimum weigh values to outstrip nonlinearities like gravity. The proposed algorithm freezes a set of adaptive weights in a feedforward-like component in the CMAC. When the feedforward component has been established, the algorithm starts to learn another set of weights which contribute to feedback-like terms in the CMAC and these weights get frozen when they no longer reduce a cost-functional This control solution based in the CMAC ANN was validated with numerical simulations to a two-link flexible-joint robot. In [28], a novel output feedback controller with a feedforward term based on the Radial Basis Function (RBF) ANN was proposed to compensate for uncertainties in the dynamic model of a robotic exoskeleton. This advanced control solution requires only position information for the RBF inputs. In [29], a PD controller with a BSNN feedforward compensation was applied to a DPR to regulate the trajectory tracking for a P&P application, demonstrating that the addition of intelligent compensation terms may reduce the tracking error considerably and might cancel the steady-state error for the PD controller. However, only the error signal was taken into consideration as inputs of the BSNN so that the resulting dynamic approximation was not accurate.

3 DPR Modeling

3.1 System Description

The DPR is a 3-DOF (Degrees of Freedom) PKM designed for P&P tasks; its mechanical structure is composed mainly of two platforms, fixed base, and traveling plate; the last one performs translational movements with a fixed orientation. The traveling plate is connected to the fixed base through three identical kinematic chains. Each kinematic chain consists of two parts, a rear-arm and a forearm, which is composed of two parallel bars, both are connected by way of passive spherical joints. The DPR rear-arms are mounted directly to the actuators located on the fixed base through rotational joints, while the forearms are connected to the traveling through a set of passive spherical joints. The dynamic model is represented in the joint space whose variables are denoted as $\mathbf{q} = [q_1 \; q_2 \; q_3]^T$ however, the position of the traveling plate is given in Cartesian coordinates as $\mathbf{X} = [x \; y \; z]^T$. The schematic diagram of the DPR is shown in Fig. 1.

3.2 Inverse Kinematic Model

Inverse Kinematic Model (IKM) for PKMs with delta-like architecture is formulated trough the Loop Closure Method [30]. Considering Fig. 1 the closed-loop equation for the DPR is established as follows:

$$||\mathbf{B}_i\mathbf{C}_i||^2 = l_i^2 \tag{1}$$

$$\mathbf{A}_i = R_b \left[\cos(\alpha_i) \;\; \sin(\alpha_i) \;\; 0\right]^T \tag{2}$$

where $\mathbf{A}_i$, $\forall i = 1...3$ represents the location of the three actuated joints expressed in the fixed reference frame. R_b is the fixed-base radius, the actuated joints are placed with the following angles $\boldsymbol{\alpha} = \left[\frac{3\pi}{2} \;\; \frac{\pi}{6} \;\; \frac{5\pi}{6}\right]^T$.
The points $\mathbf{B}_i$ and $\mathbf{C}_i$ whose coordinates are expressed in the fixed reference frame $O - x_o, y_o, z_o$ are defined as follows:

$$\mathbf{B}_i = \mathbf{A}_i + L\left[\cos(\alpha_i)\cos(q_i) \;\; \sin(\alpha_i)\cos(q_i) \;\; -\sin(q_i)\right]^T \tag{3}$$

$$\mathbf{C}_i = \left[R_p\cos(\alpha_i) + x \;\; R_p\sin(\alpha_i) + y \;\; z\right]^T \tag{4}$$

being L the arm length and R_p is the traveling-plate radius. An auxiliary frame located at $\mathbf{A}_i$-x_i, y_i, z_i is defined, where

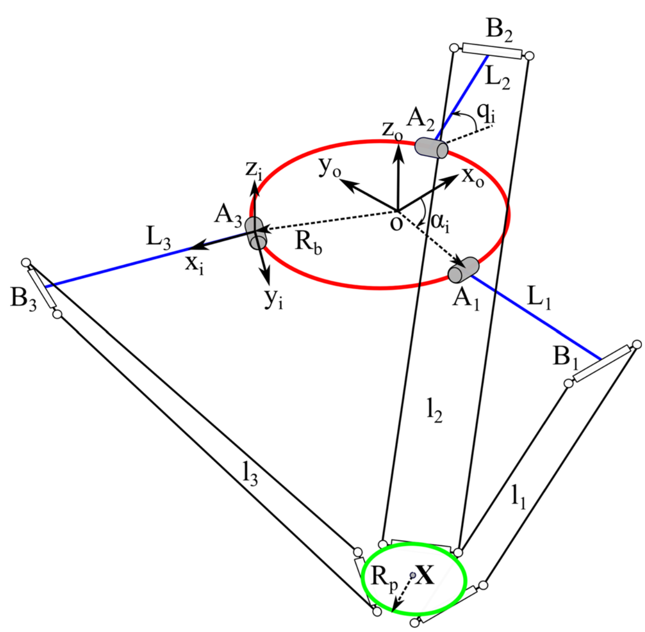


Fig. 1 Illustration of a DPR kinematic chain

the auxiliary vectors ${}^{i}\mathbf{x}_i$ and ${}^{i}\mathbf{y}_i$ are defined as:

$$ {}^{i}\mathbf{x}_i = \left[\cos(\alpha_i)\ \sin(\alpha_i)\ 0\right]^T \tag{5}$$

$$ {}^{i}\mathbf{y}_i = \left[-\sin(\alpha_i)\ \cos(\alpha_i)\ 0\right]^T \tag{6}$$

Having defined all the equations that involve the closed-loop equation the expression (1) is re-write in the following form to obtain the values of q_i.

$$ D_i \sin(q_i) + E_i \cos(q_i) + F_i = 0 \quad \forall i = 1, 2, 3 \tag{7}$$

where $D_i = 2L_i(\mathbf{A}_i\mathbf{C}_i \cdot \mathbf{z}_o)$, $E_i = 2L_i(\mathbf{A}_i\mathbf{C}_i \cdot {}^{i}\mathbf{x}_i)$, and $F_i = l_i^2 - L_i^2 - ||\mathbf{A}_i\mathbf{C}_i||^2$. Solving (7) the values of q_i can be obtained using the following expression:

$$ q_i = \arctan\left(\frac{-D_i \pm \sqrt{\Delta_i}}{F_i - E_i}\right) \tag{8}$$

Being Eq. 8 the corresponding IKM for the DPR, with $\Delta_i = D_i^2 + E_i^2 - F_i^2$.

3.3 Inverse Dynamic Model

The Inverse Dynamic Model (IDM) for the DPR has been developed considering the methodology presented in [20]. For PKMs with delta-like architecture, some simplifications to develop their dynamic model are considered, these simplifications are discussed in more detail in [30] and [31]. The simplifications are the following:

- Since obtaining an accurate frictional model for PKMs, the frictional forces dry and viscous are omitted in the analysis.
- The rotational inertia of the forearms is neglected. Nevertheless, its mass is divided into two equivalent parts; one part is added to the rear-arm mass, and the other part is joined to the traveling plate mass. This simplification is justified if the mass of the forearms is smaller than the other components of the robot.

We can establish the inverse dynamic model in function of the torques produced by the actuators $\mathbf{\Gamma}_{act} \in \mathbb{R}^{3\times1}$, the rear-arms with a half mass of the forearms $\mathbf{\Gamma}_{rf} \in \mathbb{R}^{3\times1}$ and, the traveling plate with the other half mass of the forearms $\mathbf{\Gamma}_{ftp} \in \mathbb{R}^{3\times1}$ as follows:

$$ \mathbf{\Gamma} = \mathbf{\Gamma}_{act} + \mathbf{\Gamma}_{rf} + \mathbf{\Gamma}_{ftp} \tag{9}$$

The produced torques owing to motor's inertia are obtained as follows:

$$ \mathbf{\Gamma}_{act} = \mathbf{I}_{act}\ddot{\mathbf{q}} \tag{10}$$

where $\mathbf{I}_{act} = diag([I_{act}]) \in \mathbb{R}^{3\times3}$ is a square diagonal matrix containing the inertia values of each motor. Considering the second simplification mentioned above, one can derive the dynamics of the rear-arms and forearms as follows. For the rear-arms torques are computed through the following equation:

$$ \mathbf{\Gamma}_{\boldsymbol{ra}}\boldsymbol{(t)} = \mathbf{I}_{ra}\ddot{\mathbf{q}} + \mathbf{M}_{ra}gL_c\cos(q) \tag{11}$$

where $\mathbf{I}_{ra} = diag([I_{ra}]) \in \mathbb{R}^{3\times3}$ is the inertia matrix of the rear-arms', $\cos(\mathbf{q})$ is a vector of 3×1, representing the cosine of each angle q_i $\forall i = 1, ..., 3$, $\mathbf{M}_{ra} = diag([m_{ra}]) \in \mathbb{R}^{3\times3}$ is the mass matrix of the rear-arms', L_c is the distance from the rotational axis of the rear-arm to its gravity center, and $\cos(q)$ is composed as follows:

$$ \cos(q) = [\cos(q_1)\ \cos(q_2)\ \cos(q_3)]^T \tag{12}$$

Considering the second simplification, one may express the torque contributions of the forearms by means the following expression:

$$ \mathbf{\Gamma}_{fa}(t) = \mathbf{I}_{fa}\ddot{\mathbf{q}} + \mathbf{M}_{fa}gLcos(q) + \mathbf{J}_{inv}^T\mathbf{M}_{nfa}(\ddot{\mathbf{X}} + \mathbf{G}) \tag{13}$$

Where $\mathbf{I}_{fa} = diag([L^2\frac{m_{fa}}{2}]) \in \mathbb{R}^{3\times3}$, $\mathbf{M}_{fa} = diag([\frac{m_{fa}}{2}]) \in \mathbb{R}^{3\times3}$,and $\mathbf{M}_{nfa} \in \mathbb{R}^{3\times3} = diag([3\frac{m_{fa}}{2}])$ where m_{fa} is the forearm mass considering the two parallel bars. $\mathbf{J}_{inv} \in \mathbb{R}^{3\times3}$ is the inverse Jacobian matrix, $\ddot{\mathbf{X}} \in \mathbb{R}^{3\times1}$ is the Cartesian acceleration vector of the traveling plate, L is the rear-arm length, and $\mathbf{G} = \left[0\ 0\ g\right]^T \in \mathbb{R}^{3\times1}$ is the gravity vector with $g = 9.81\ m/s^2$. Applying the Newton-Euler equation to the traveling plate we obtain the following expression:

$$ \mathbf{F}_p = \mathbf{G}_p \tag{14}$$

where $\mathbf{F}_p$ and $\mathbf{G}_p$ are the inertial and gravity forces acting on the traveling plate represented in the following expressions:

$$ \mathbf{F}_p = \mathbf{M}_p\ddot{\mathbf{X}} \tag{15}$$

$$ \mathbf{G}_p = -\mathbf{M}_p\mathbf{G} \tag{16}$$

being $\ddot{\mathbf{X}} \in \mathbb{R}^{3\times1}$ the Cartesian acceleration vector. The mass matrix of the traveling plate is composed as follows:

$$ \mathbf{M}_p = diag([m_p\ m_p\ m_p]) \tag{17}$$

where m_p is the traveling plate mass. The inverse Jacobian matrix is used to compute the traveling plate torque contributions produced by the inertial forces and gravity force as follows:

$$ \mathbf{\Gamma}_{tp} = \mathbf{J}_{inv}^T\mathbf{M}_p(\ddot{\mathbf{X}} + G) \tag{18}$$

The dynamic equation of the forearms (13) should be split into two parts, one part is added to Eq. 18, and the other part is added to Eq. 18 to obtain $\mathbf{\Gamma}_{rf}$ and $\mathbf{\Gamma}_{ftp}$. The torque contributions due to the rear-arms and the half mass of the forearms are given as follows:

$$ \mathbf{\Gamma}_{rf} = \mathbf{I}_{rf}\ddot{\mathbf{q}} + \mathbf{M}_{rf}g\mathbf{cos(q)} \tag{19}$$

Where $\mathbf{I}_{rf} \in \mathbb{R}^{3\times3}$ is a square diagonal matrix whose elements are formed by: $I_{rf} = I_{ra} + L^2\frac{m_{fa}}{2}$. The resulting mass matrix is expressed as:

$$\mathbf{M}_{rf} = diag([m_{rf}\ \ m_{rf}\ \ m_{rf}]) \tag{20}$$

With $m_{rf} = m_{ra}L_c + \frac{m_{fa}L}{2}$. To express the inverse dynamic model in function of the joint space variables, it is essential to take into consideration the following relations based on the inverse Jacobian matrix:

$$\dot{\mathbf{X}} = \mathbf{J}_{inv}\dot{\mathbf{q}} \tag{21}$$

$$\ddot{\mathbf{X}} = \mathbf{J}_{inv}\ddot{\mathbf{q}} + \dot{\mathbf{J}}_{inv}\dot{\mathbf{q}} \tag{22}$$

Substituting Eqs. 18, 19, and 10 in Eq. 9 and taking into account (12) we state the inverse dynamic model as follows:

$$\mathbf{M(q)\ddot{q}} + \mathbf{C(q, \dot{q})\dot{q}} + \mathbf{G(q)} = \mathbf{\Gamma} \tag{23}$$

where:

- $\mathbf{M(q)} = \mathbf{I}_{act} + \mathbf{I}_{rf} + \mathbf{J}_{inv}^T\mathbf{M}_p\mathbf{J}_{inv}$
- $\mathbf{C(q, \dot{q})} = \mathbf{J}_{inv}^T\mathbf{M}_p\dot{\mathbf{J}}_{inv}$
- $\mathbf{G(q)} = (\mathbf{M}_{rf}\mathbf{cos(q)} + \mathbf{J}_{inv}^T\mathbf{M}_p)\mathbf{G}$

The kinematic and dynamic parameters of the DPR are shown in Tables 1 and 2 respectively.

4 Control Strategy

The main objective of a DPR is to perform high speed and high accuracy P&P operations with the smallest possible tracking error. To reach this objective, it is crucially essential to design a control scheme capable of keeping the precision under abrupt changes of mass and acceleration. To satisfy these demands, we propose integrating the RISE control algorithm with an adaptive feedforward compensation term. The main feature of RISE controller can ensure semi-global asymptotic stability in the presence of general uncertain disturbances [32]. It is well known in robotics control that the addition of a feedforward term can compensate the inherent nonlinearities and improve the system performance. However, sometimes, the dynamic model or dynamic parameters as masses and inertia are unknown or not measurable. Consequently, wrong parameter estimation or an inaccurate dynamic model can harm the efficiency of the control scheme instead of improving. ANNs are an attractive solution for nonlinear modeling systems due to their ability to identify unknown dynamic models through a set of inputs and outputs related to each other. BSNN is a kind of ANN formed by three parts: A lattice used to normalize the inputs, a single layer set of basis functions defined over the lattice, and the network output, which is a linear combination of the basis functions with the adjustable weights [33]. This ANN is very suitable for nonlinear model identification in real-time due to its construction formed by only one hidden layer of basis functions avoiding large number calculus compared to any multilayer ANN. In this work, we employed BSNNs to approximate the inverse dynamics for each kinematic chain of the DPR. Having in mind the benefits of RISE control and ANN, we establish the following control scheme for the DPR:

Table 1 Summary of the DPR kinematic parameters

Parameter	Description	Value
L	Rear-arm length	0.3 m
l	Forearm length	0.624 m
R_b	Base platform radio	0.1267 m
R_p	Traveling plate radio	0.0497 m

Table 2 Summary of the DPR dynamic parameters

Parameter	Description	Value
m_{tp}	Mobile platform mass	0.19 Kg
m_{ra}	Rear-arm mass	0.29 Kg
m_{fa}	Forearm mass	0.28 Kg
I_{ra}	Rear-arm inertia	0.0213 Kgm^2
I_{act}	Motor inertia	3.8 $\times 10^{-6}$ Kgm^2

$$\mathbf{\Gamma} = \mathbf{\Gamma}_{RISE} + \hat{\Sigma}(\mathbf{q_d}, \dot{\mathbf{q_d}}, \ddot{\mathbf{q}}_\mathbf{d}, \mathbf{e_1}) \tag{24}$$

where $\mathbf{\Gamma}_{RISE} \in \mathbb{R}^{3\times1}$ corresponds to feedback RISE feedback control and the term $\hat{\boldsymbol{\Sigma}}(\boldsymbol{q_d}, \dot{\boldsymbol{q}}_{\boldsymbol{d}}, \ddot{\boldsymbol{q}}_{\boldsymbol{d}}, \boldsymbol{e_1}) \in \mathbb{R}^{3\times1}$ is the intelligent vector-based term on BSNNs. Figure 2 illustrates a general overview of the proposed control technique.

The position tracking error in joint space $\mathbf{e}_q(t) \in \mathbb{R}^{3\times1}$, is defined as:

$$\mathbf{e}_q = \mathbf{q}_d - \mathbf{q} \tag{25}$$

where $\mathbf{q}_d$ is the desired joint position and $\mathbf{q}$ is the actual joint position. RISE control requires the evaluation of the combined filtered tracking error in joint space denoted by the following expression:

$$\mathbf{e}_1 = \dot{\mathbf{e}}_q + \boldsymbol{\alpha}_1\mathbf{e}_q \tag{26}$$

where $\boldsymbol{\alpha}_1, \in \mathbb{R}^{3\times3}$ is a positive-definite, diagonal matrix. The RISE feedback control expression is defined by the following equation:

$$\begin{aligned}\mathbf{\Gamma}_{RISE} &= (\mathbf{K}_s + \mathbf{I})\mathbf{e}_1(t) - (\mathbf{K}_s + \mathbf{I})\mathbf{e}_1(t_0) + \\ &\int_0^t [(\mathbf{K}_s + \mathbf{I})\boldsymbol{\alpha}_2\mathbf{e}_1(\tau) + \boldsymbol{\beta}\mathbf{sgn}(\mathbf{e}_1(\tau))]d\tau\end{aligned} \tag{27}$$

where $\mathbf{K}_s$, $\boldsymbol{\alpha}_2$, $\boldsymbol{\beta} \in \mathbb{R}^{3\times3}$ are positive-definite, diagonal matrices, $\mathbf{I} \in \mathbb{R}^{4\times4}$ is the identity matrix, and $\mathbf{sgn}(.)$ is the vector of the sign functions of the first filtered tracking error. The term $(\mathbf{K}_s + \mathbf{I})\mathbf{e}_1(t_0)$ is used to ensure a zero initial

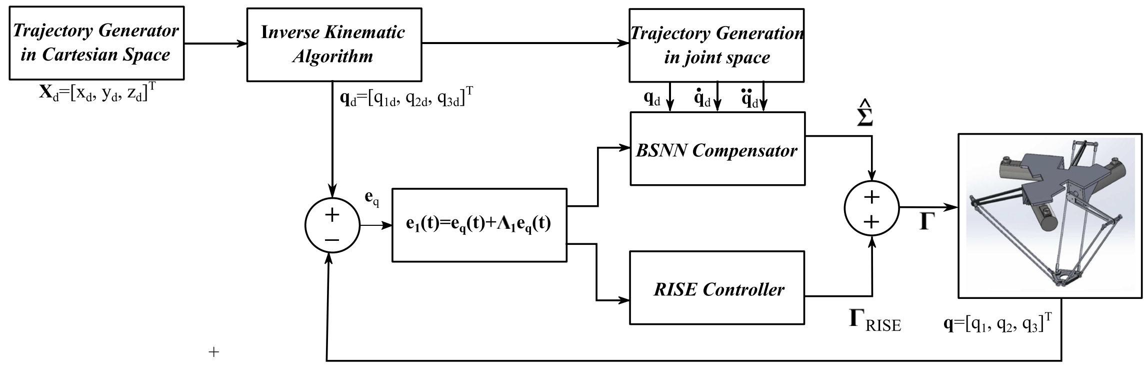


Fig. 2 Representation of the proposed control scheme with BSNN compensation for the DPR

control input at $t = 0$. The vector containing the BSNNs outputs is defined as:

$$\hat{\boldsymbol{\Sigma}} = [\hat{\sigma}_1 \ \hat{\sigma}_2 \ \hat{\sigma}_3]^T \tag{28}$$

where $\hat{\sigma}_i \quad \forall i = 1, 2, 3$ denotes the respective BSNN output used to approximate the dynamics of one DPR kinematic chain.

4.1 Design of the Feedfoward Term Based on BSNNs

As it was mentioned above, the BSNNs aims to estimate on-line the dynamic behavior of the DPR to include it into the control loop as a feedforward compensation term. In robotics the feedforward control is represented by the following expression:

$$\mathbf{M}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}_d, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + \mathbf{G}(\mathbf{q}_d) = \boldsymbol{\Gamma}_{FW} \tag{29}$$

However, for the proposed control scheme $\mathbf{M} \in \mathbb{R}^{3\times3}$, $\mathbf{C} \in \mathbb{R}^{3\times3}$, $\mathbf{G} \in \mathbb{R}^{3\times1}$ are considered unknown. One can see that the Inertia, Centripetal/Coriolis matrices, and the gravity vector are evaluated with the desired trajectories $\mathbf{q}_d$, $\dot{\mathbf{q}}_d$, $\ddot{\mathbf{q}}_d$. Therefore, we set the trajectories values as the data input for the BSNNs. An important aspect of the design of each BSNN is to define the input space lattice formed by a set of n knot-vectors, one-knot vector for each input axis. Once the input data is established, the next step is to define the K order, shape, number, and distribution of the basis functions. The K order defines the shape of the basis functions, i.e., if $K = 1$, we obtain piecewise constant functions, $K = 2$ leads to piecewise linear functions, $K = 3$ generates piecewise quadratic functions and, when $K = 4$ piecewise cubic functions are obtained. Selecting a higher-order for the functions result in a better approximation. The number of knots and the value of each one, as well as the interval between them, are set by prior knowledge of the selected BSNN inputs. Dynamics of Parallel Robots are highly and complex; thus, we selected basis functions of third-order to acquire an accurate approximation of the dynamics behavior without making a greater number of calculations as may occur with cubic functions. A knot-vector is defined for each input axis considering the extreme admissible values of the trajectories as the maximum and minimum values of the input vectors. For the axes where $\mathbf{q}_d$ are the inputs the minimum and maximum values are from -1 to 1 rad respectively, -10 to 10 rad/s for $\dot{\mathbf{q}}_d$ and -200 to 200 rad/s^2 for $\ddot{\mathbf{q}}_d$. Once the input range is already defined for the input axes, the next step is to define the number and distribution of $j-th$ knots of the vector. Each knot-vector is formed by 8 knot-points and they are distributed in groups of four elements to generate three b-spline functions that share some knot-points among them. We selected this configuration because it gives a good approximation of the system behavior, as being reported in the results section. In Fig. 3, the distribution of the knot-points and B-spline functions for each input axis are depicted. The knot-points values for the input axes are given in Table 3.

We proceeded to present the expression of univariate B-Spline basis function, which is defined through the following recurrence relationship [34]:

$$\begin{aligned} S_K^j(u) &= \left(\frac{u-\lambda_{j-K}}{\lambda_{j-1}-\lambda_{j-K}}\right) S_{K-1}^{j-1}(u) + \left(\frac{\lambda_j - u}{\lambda_j - \lambda_{j-K+1}}\right) S_{K-1}^{j}(u) \\ S_1^j(u) &= \begin{cases} 1 & \text{if } u \in I_j \\ 0 & \text{other cases} \end{cases} \end{aligned} \tag{30}$$

where u corresponds to the input, λ_j is the jth knot point and $I_j = [\lambda_{j-1}, \quad \lambda_j)$ is the jth interval between two-knot points, and K is the order of the output function. The output of each one of the BSNN can be written as follows [35]:

$$\hat{\sigma}_i = \sum_{m=1}^{P} a_m w_m = \mathbf{a}_i^T \mathbf{w}_i \quad \forall i = 1, 2, 3 \tag{31}$$

where $\mathbf{a}_i$ is a P-dimensional vector which contains the outputs of the BSNN basis functions and, $\mathbf{w}_i$ is the weights vector. The diagram depicted in Fig. 4 represents the BSNN configuration for the DPR dynamic estimation.

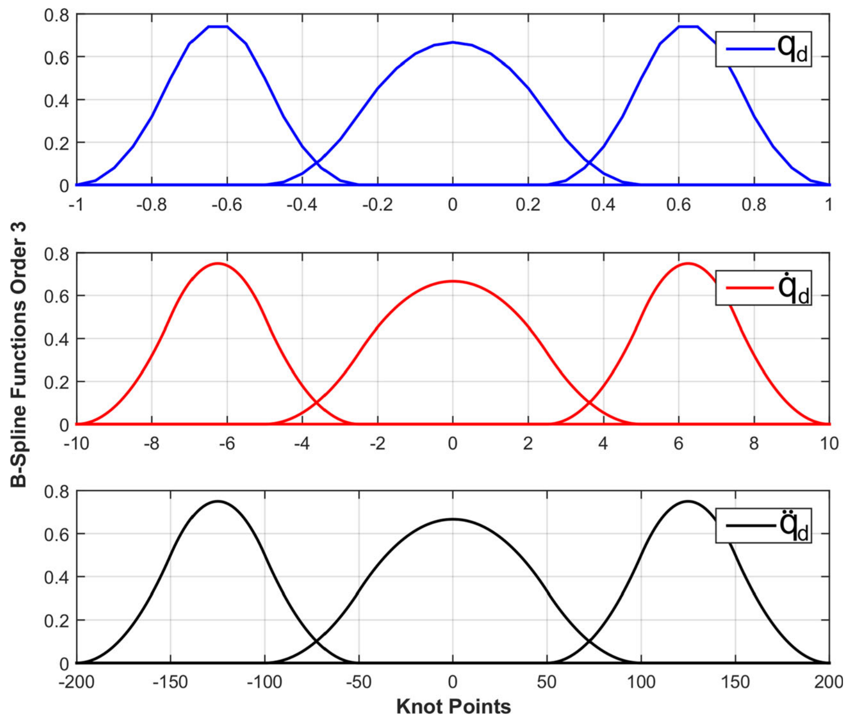


Fig. 3 Distribution of the proposed activation functions of order 3 for the respective inputs

4.2 Training Algorithm

An instantaneous training algorithm is used for the BSNN; this algorithm only adjusts the weights corresponding to the active basis functions. The instantaneous learning rule is formulated, minimizing an instantaneous estimation of a performance function of the Mean Square Error (MSE) of the output, and the parameters are updated using descending gradient rules. The MSE estimate is given by:

$$J(t) = (\hat{\sigma}(t) - \sigma(t))^2 \tag{32}$$

A variation of the standard descending gradient is the Normalized Least Mean Square (NLMS) algorithm employed for instantaneous training. We used this formulation as a learning rule because it uses few computational resources, which is essential for real-time implementation. The learning rule is given as follows [35]:

$$\mathbf{W}_i = \mathbf{W}_i(t-1) + \frac{\gamma \tilde{\sigma}_i(t)}{||\mathbf{a}_i(t)||_2^2}\mathbf{a}_i(t) \quad \forall i = 1, 2, 3 \tag{33}$$

Table 3 Knot-points Vectors' distribution

Input	Knot-points Vector
$\mathbf{q}_d$	$[-1\ -0.75\ -0.5\ -0.25]$ $[-0.5\ -0.25\ 0.25\ 0.5]$ $[0.25\ 0.5\ 0.75\ 1]$
$\dot{\mathbf{q}}_d$	$[-10\ -7.5\ -5\ -2.5]$ $[-5\ -2.5\ 2.5\ 5]$ $[2.5\ 5\ 7.5\ 10]$
$\ddot{\mathbf{q}}_d$	$[-200\ -150\ -100\ -50]$ $[-100\ -50\ 50\ 100]$ $[50\ 100\ 150\ 200]$

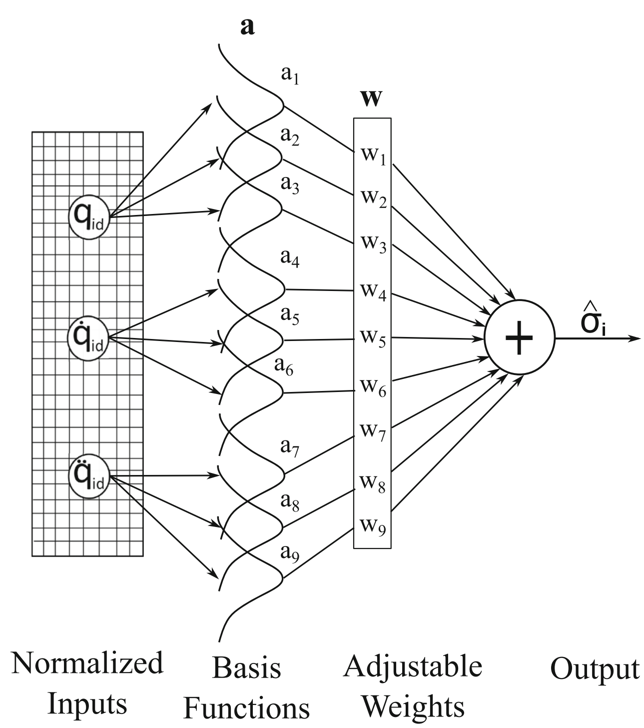


Fig. 4 Diagram of the proposed BSNN used as a compensation term for each kinematic chain of the DPR

where γ is the learning rate, $\mathbf{a}_i$ is the vector that contains the output of the basis functions, $\mathbf{W}_i$ is the adjustable weights vector, and $\tilde{\sigma}_i(t) = \sigma_i(t) - \hat{\sigma}_i(t)$ is the BSNN output error. To do the on-line training of the BSNN, it is necessary an error signal that is the difference between the real variable and the estimated by the BSNN. However, in this case, the real value is not available since it is required to obtain through the BSNN. For this reason, it is consistent with using the measurement of the robot's position and comparing it with the values of the established desired trajectory to obtain an error signal. In this case, $\tilde{\sigma}_i(t)$ is estimated using the composed tracking error $\mathbf{e}_1$, as illustrated in Fig 2.

5 Simulation and Results

The performance of the proposed control scheme is compared to the standard RISE controller under different scenarios for two case studies. The first one consists of a high-speed P&P trajectory task, and the second one is a spiral trajectory tracking evaluated under different speeds. The performance of each control scheme is quantified using the Root Mean Square Error (RMSE) formula. The following two equations established the RMSE in Cartesian and joint space form respectively:

$$RMSE_C = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(e_x^2(k) + e_y^2(k) + e_z^2(k))} \tag{34}$$

$$RMSE_J = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(e_{q1}^2(k) + e_{q2}^2(k) + e_{q3}^2(k)} \tag{35}$$

where e_x, e_y, e_z denote the Cartesian position tracking error of the traveling plate along the x, y, z axes, while e_{q1}, e_{q2}, e_{q3} are the different joint space tracking errors. Moreover, N is the number of samples and k the current sample. The controller parameters for RISE and RISE BSNN are shown in Table 4.

Table 4 Controllers parameters RISE/RISE BSNN

Parameter	Value
α_1	110
α_2	8
Ks	60
β	3
γ	0.53

5.1 Case Study 1

The P&P trajectory used for this case study is represented in Cartesian space by Fig. 5, and it composes of two illustrations. The left illustration represents the tracking trajectory for the first scenario executed by the DPR without any payload, while in the second scenario, the DPR moves masses of 1 Kg along trajectory sections. The sections of the trajectory where the traveling plate of the DPR moves a mass are depicted with a dotted line in red color, whereas the solid lines in blue are the sections of the trajectory where the DPR is moving without any payload. This trajectory is generated using the polynomial interpolation of fifth-order [36], [37]. This polynomial function is generated thanks to the following two expressions:

$$x_f = x_i + r(t)\Delta x, \qquad for \quad 0 \le t \le t_f \tag{36}$$

And:

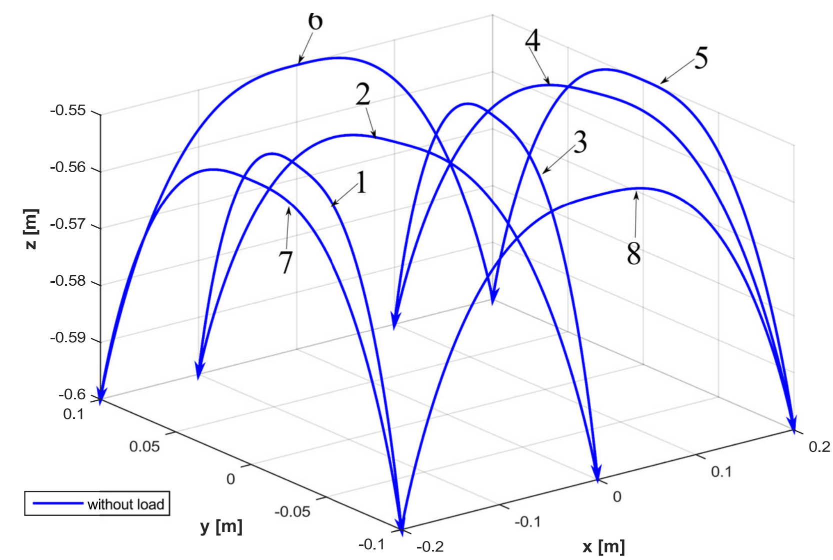
$$r(t) = 10\left(\frac{t}{t_f}\right)^3 - 15\left(\frac{t}{t_f}\right)^4 + 6\left(\frac{t}{t_f}\right)^5 \tag{37}$$

where x_i is the initial position, x_f is the final position; both are given in Cartesian space, $r(t)$ is the trajectory function of two points, $\Delta_x = x_f - x_i$, and t_f is the duration of the movement. The desired trajectories respect to time in Cartesian space are generated through Eqs. 36 and 37, they are represented in Fig. 6. The sequence of movements for the P&P trajectory in the *(x,y)* plane is the following.

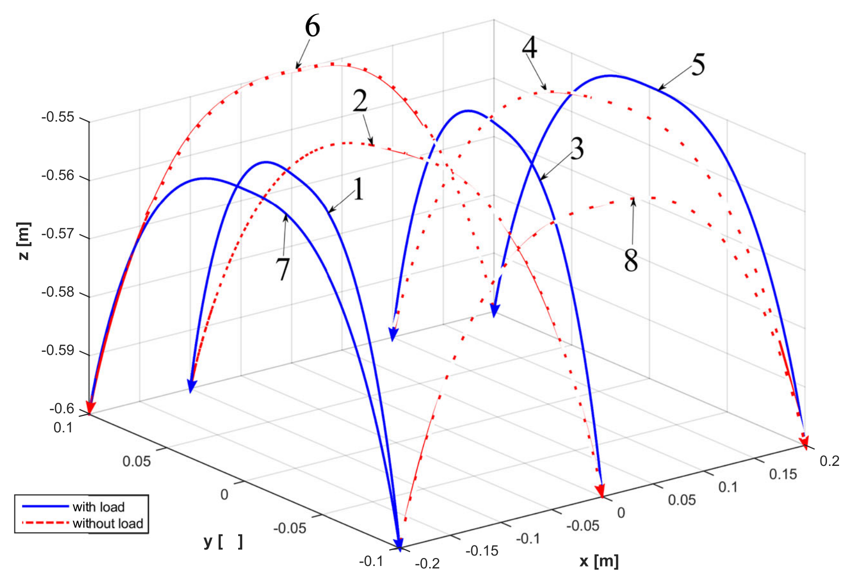
1. Start-Pick: from (-0.2,-0.1) to (-0.1,0.1).
2. Pick-Place: from (-0.1,0.1) to (0,-0.1).
3. Place-Pick: from (0,-0.1) to (0.1,0.1).
4. Pick-Place: from (0.1,0.1) to (0.2,-0.1).
5. Place-Pick: from (0.2,-0.1) to (0.2,0.1).
6. Pick-Place: from (0.2,0.1) to (-0.2,0.1).
7. Place-Pick: from (-0.2,0.1) to (-0.2,-0.1).
8. Pick-Place: from (-0.2,-0.1) to (0.2,-0.1).

The previous movement sequences are performed in 0.3 seconds for both scenarios. The simulation results for the first scenario are presented in Figs. 7 and 8. Figure 7 shows the tracking error graphs in Cartesian and joint space. As it can be noted, the tracking errors of RISE BSNN are noticeably smaller than those of Standard RISE control due to the BSNN compensation terms reducing the effect of nonlinearities, resulting in a better tracking performance. Figure 8 displays the generated torques by the Standard RISE and our proposed RISE BSNN in the first column graphs, whereas the control signals that form our proposed controller (i.e., RISE contribution and BSNN contribution) are in the second column. It is noteworthy that the behavior of the BSNNs outputs is very similar to the torques produced for both control schemes, this is due to the accurate approximation of the inverse dynamic

Fig. 5 Desired 3D trajectory for a P&P Task. The lines in red correspond to trajectory portions where the DPR is moving with a payload and the blue lines are the corresponding portions without payload



(a) Desired trajectory in Cartesian space for scenario 1 case study 1



(b) Desired trajectory in Cartesian space for scenario 2 case study 1

model of DPR computed by the BSNNs. Moreover, as can be seen in the same figure, the BSNN control term produces most of the torque required to reach the desired position, this is due to its good approximation of the inverse dynamic model for the DPR and, on the other hand, the term corresponding to the RISE control produces the extra torque needed to achieve the desired position accurately. The obtained tracking errors for the second scenario are displayed in the graphs of Fig. 9. It can be appreciated that the amplitude of tracking errors has increased for the two controllers as a consequence of the addition of the moving mass. However, the RISE BSNN control law's performance is still widely better than the Standard RISE controller. The values of produced torques of this second scenario and the contribution signals of the RISE BSNN controller are exposed in Fig. 10. It can be seen that the curves have doubled compared to control signals for scenario owing to both controllers requiring more energy to move the payload from one point to another. Figure 11 shows the evolution of the BSNNs' adaptive weighs for each kinematic chain of the DPR for the two scenarios. It can be seen in all cases that the initial value of the weights is zero, and as the trajectories are executed, not all the weights evolve together; this is because of the BSNNs update only the associate weights to the current input values of the BSNNs. Besides, as it can be observed, some of the adaptive weights associated with

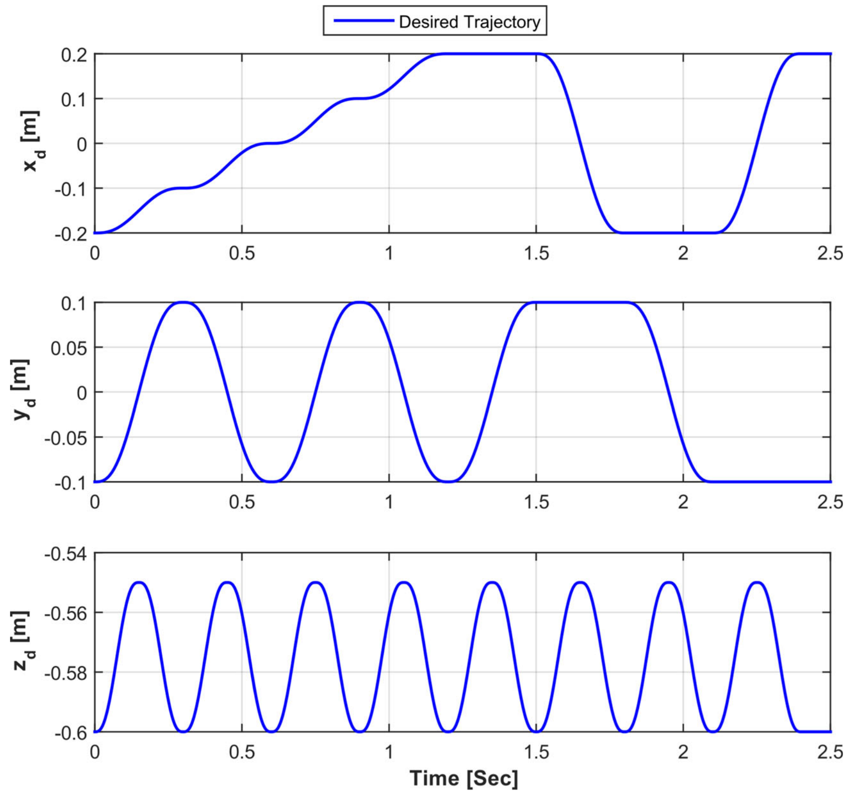


Fig. 6 Evolution of the desired trajectories in Cartesian space versus time for case study 1

extreme input values always remain zero; this is because the desired trajectories used as inputs to the BSNNs are not at those extreme range values. For example, for the case study 2 where a change in the speed was tested, for the lower speed scenario, only the weights related to the position are updated because the desired trajectory reaches the limits of the cartesian space, i.e., the main requirement for the task is only the position. In the same way, for the medium speed scenario, the related weights to the speed are now updated, too, due to the speed requirement. Finally, for the high-speed scenario, the associated weights are updated now due to the acceleration requirement. Table 5

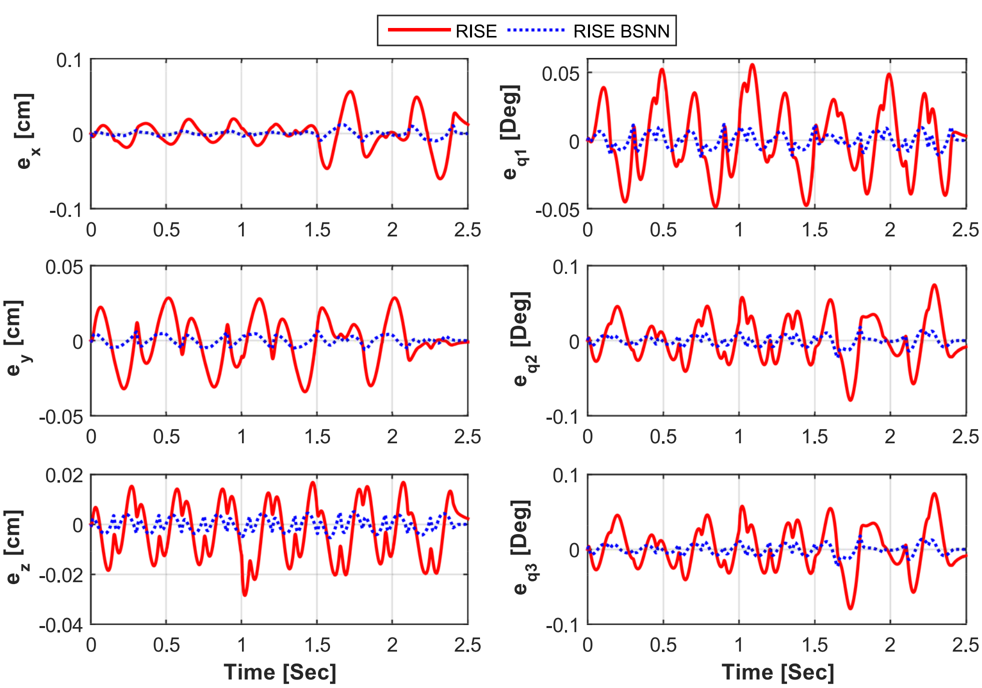


Fig. 7 Evolution of the tracking errors versus time in Cartesian and joint space for scenario 1 case study 1

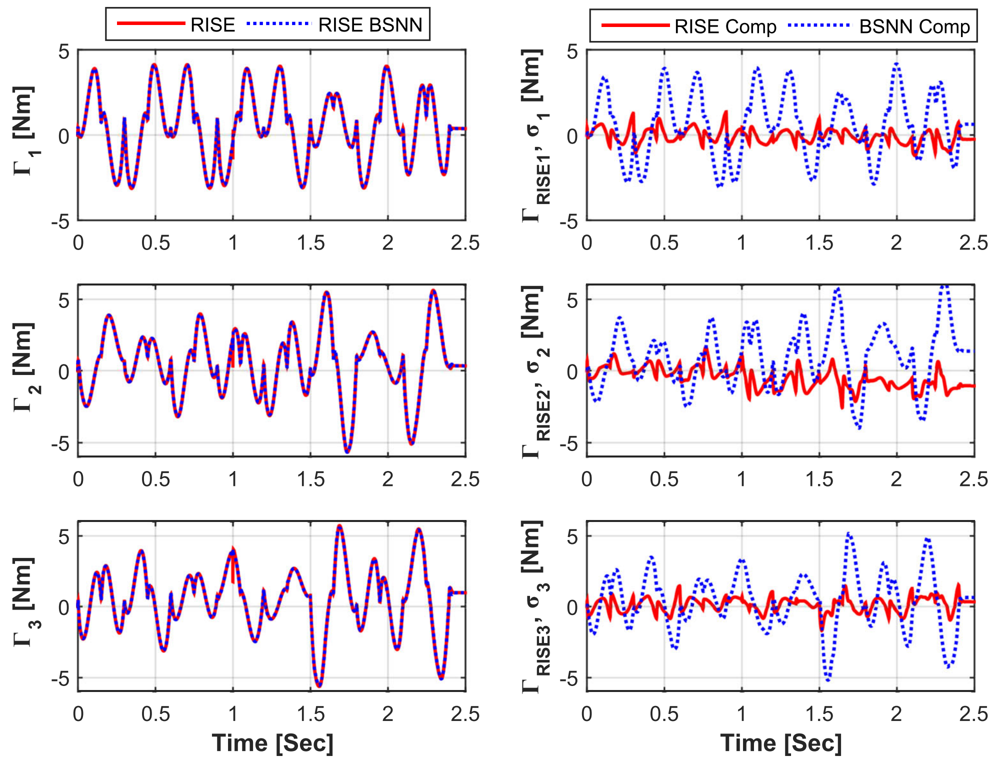


Fig. 8 Evolution of the control signals generated by RISE and RISE BSNN controllers (first column), and the control contributions of RISE BSNN (second column) versus time for scenario 1 case study 1

summarizes the performance of both controllers of the proposed two scenarios using the RMSE formulas; as it can be seen, the enhancement of RISE BSNN respect to Standard RISE is over 80% and 79% for Cartesian and joint space, respectively in two scenarios, reinforcing the presented results in Figs. 7 and 9.

5.2 Case Study 2

The desired trajectory for this case study is a spiral path on the plane (x,y) (see Fig. 12). The three scenarios proposed for this case study are subject to changes in the speed execution of the trajectory (low, medium, and high). The

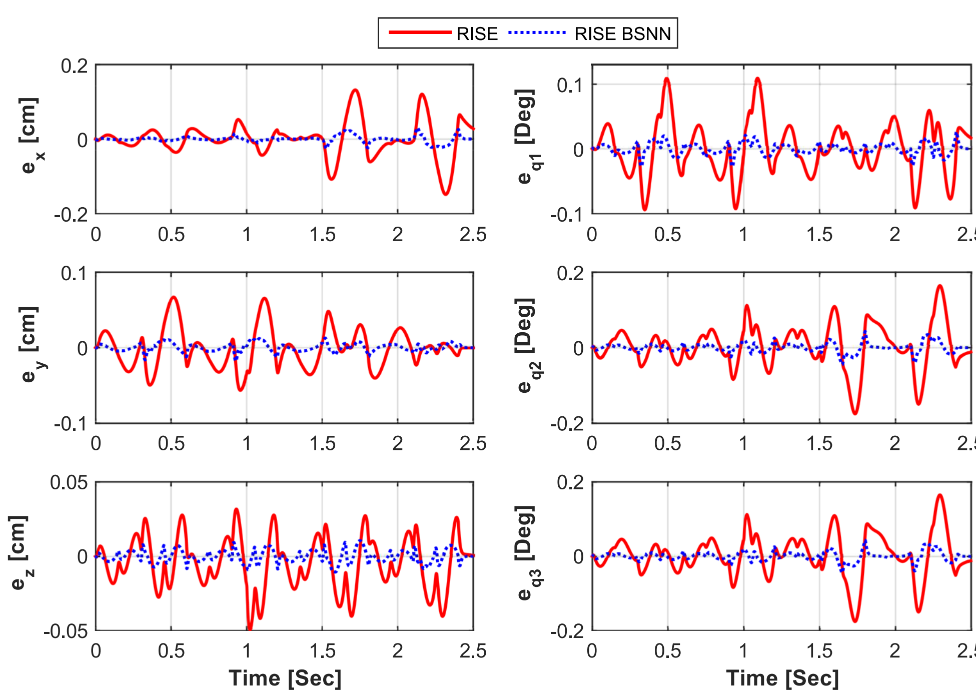


Fig. 9 Evolution of the tracking errors versus time in Cartesian and joint space for scenario 2 case study 1

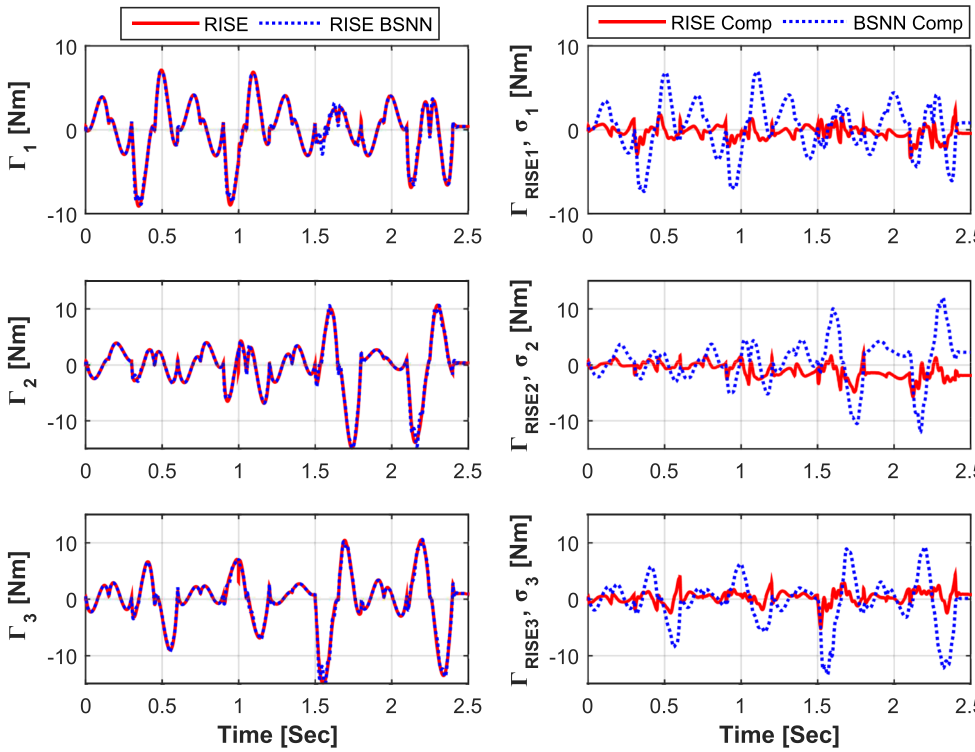


Fig. 10 Evolution of the control signals generated by RISE and RISE BSNN controllers (first column), and the control contributions of RISE BSNN (second column) versus time for scenario 2 case study 1

following equations are used to generate the desired spiral trajectory:

$$x_d = r\cos(2\pi f t)$$
$$y_d = r\sin(2\pi f t) \quad (38)$$
$$z_d = -0.6$$

$$r = 0.04 f t \quad (39)$$

where r denotes the separation distance between circular turns and f is the frequency of the circular movements. The speed changes are achieved by modifying the value of f, we define:

- $f = 0.33Hz$ for low speed
- $f = 1.75Hz$ for medium speed
- $f = 3.5Hz$ for high speed

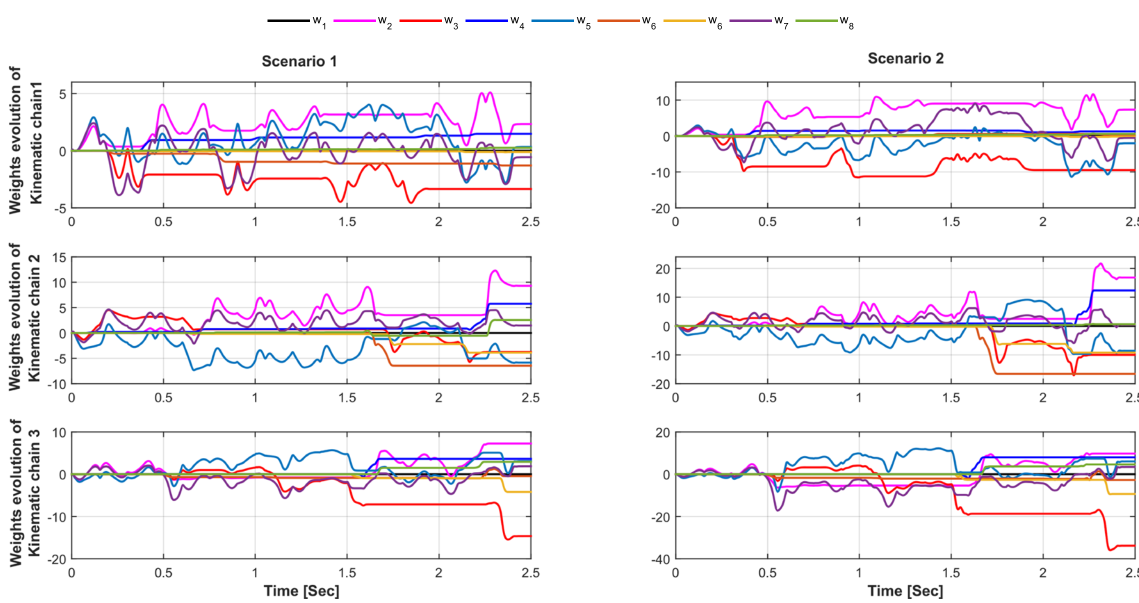


Fig. 11 Evolution of the BSNNs' weights of case study 1 for scenarios 1 and 2

Table 5 Controllers performance evaluation case study 1

Scenario	Controller	**RMSE$_C$ [cm]**	**RMSE$_J$ [Deg]**
Scenario 1	RISE	0.0285	0.0491
	RISE BSNN	0.0055	0.0102
	Enhancement	80.6%	79.1%
Scenario 2	RISE	0.0571	0.0929
	RISE BSNN	0.0109	0.0194
	Enhancement	80.9%	79.1%

The initial and final positions of the spiral trajectory given in Cartesian coordinates are (0,0,-0.6) and (0,0.2,-0.6). The objective of this study case is to know how much the changes in speed affect the controllers' performance. The obtained results from this case study are illustrated in Figs. 13, 14, 15, 16, 17, 18, 19 and 20. The tracking errors in Cartesian and Joint space are exhibits in Figs. 13, 15, and 17 for the three scenarios. As it can be noticed, as the speed is increasing, the overshoots amplitude on the tracking error signals also increases. Nevertheless, the tracking errors of the proposed controller always remain lower than the standard RISE controller. The spiral trajectory is expected to be completed in 14.8 s, 2.85 s, and 1.42 s for scenarios 1, 2, and 3, respectively. The produced torques of both controllers and the control signals of the RISE BSNN are presented in Figs. 14, 16, and 18. It is possible to see that when the speed increases, also the amplitude of the computed control signals increase. However, as in the previous case study for RISE BSNN, the control actions of the BSNNs contribute in a more significant proportion than the RISE contribution. Figure 19 presents the weighs evolution respect to time for the three scenarios (low, medium, and high speed) of this case study. As can be noted in the graphs, all the weights values are initialized in zero. In low speed, we can see that only four weights are changed along the trajectories owing to the input values of the desired trajectories stay in the range values of only one basis function; unlike in high speed where all weights are in involved since the desired trajectories reach the maximum limits of the knot-points distribution. Table 6 presents the comparison of different RMSEs for the three scenarios reinforcing the advantages of our proposed control solution. In all scenarios of this case study, the improvement of our controller compared to Standard RISE is between 60% and 80%. To have a better comprehension of how great the deterioration of the control schemes as the speed increases is, the RMSE is plotted in Fig. 20.

To justify the presented simulation results, in the previous graphs it can be seen a comparison between the tracking errors of RISE and RISE BSNN in all case studies and scenarios that the signals of the RISE BSNN errors are considerably smaller than those produced by standard RISE control. Since the learning rule of the BSNN minimizes an error signal provided by the composed tracking error to estimate on-line the dynamic behavior of the modeled system, it may be concluded that if the resulting tracking error of the RISE BSNN is smaller than produced by standard RISE, so that, the BSNN approximation is reasonably accurate. One of the most critical things in the design of the BSNN feedforward term is the selection and distribution of the knot-points. However, there are no specific criteria for the selection of these parameters, and everything depends on the prior knowledge of the system to be approximated by the designer. If the BSNNs are not properly configured, the obtained signal will deteriorate the controller performance instead of being improved. The other problem is related to the learning rule that is based-on gradient descend rules; these kinds of rules may fall in local minima problems [38].

5.3 Comparison of BSNN Compensation Against Nominal Feedforward

In the previous case studies, our proposed RISE with BSNN compensation was evaluated to standard RISE control, and

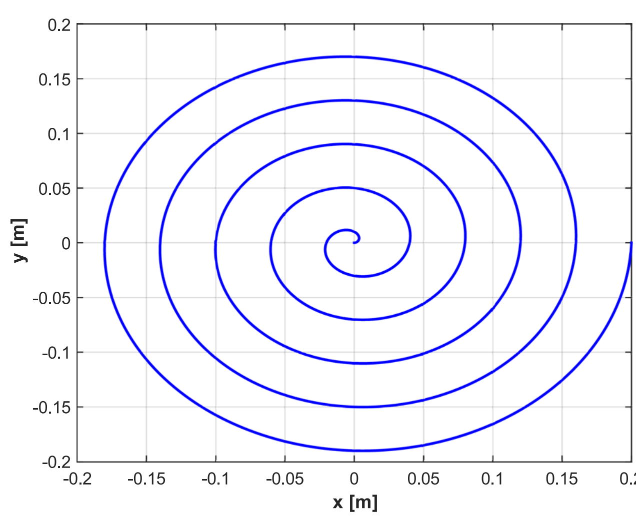


Fig. 12 Desired spiral trajectory in the plane (x,y) for case study 2

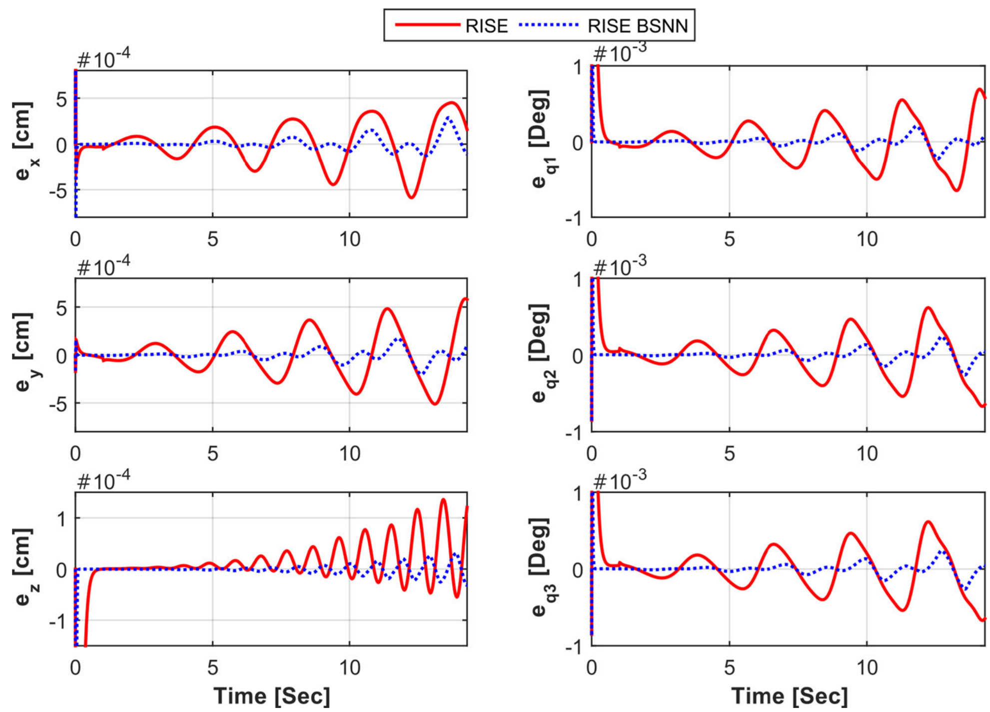


Fig. 13 Evolution of the tracking errors versus time in Cartesian and joint space corresponding to low speed for case study 2

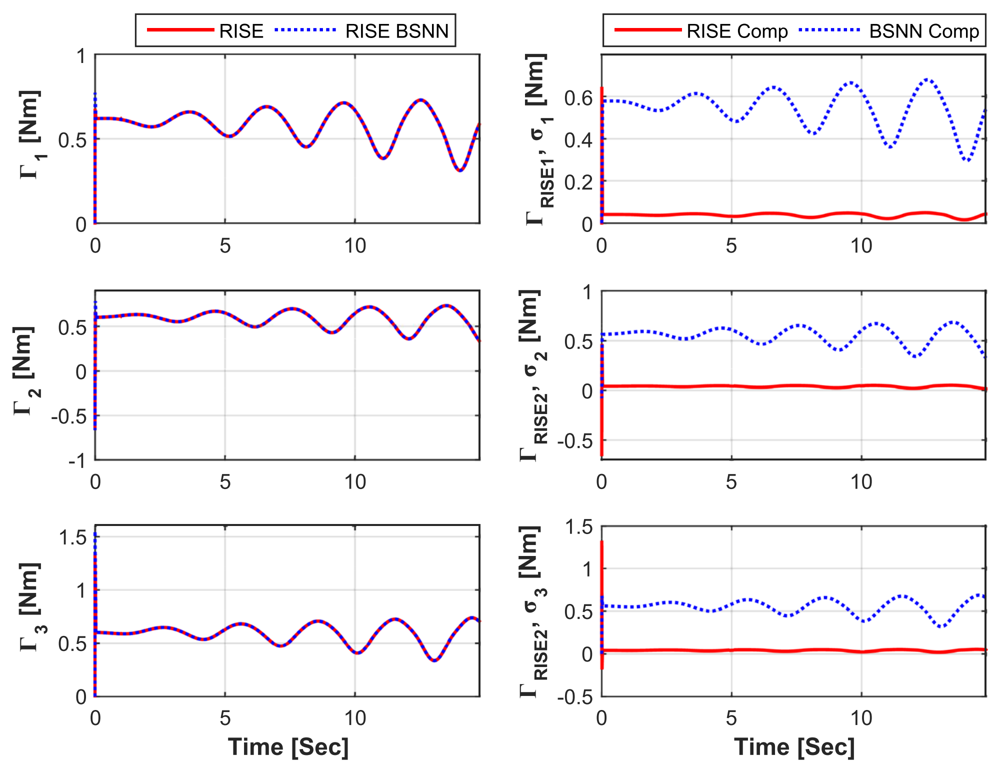


Fig. 14 Evolution of the control signals generated by RISE and RISE BSNN controllers (first column), and the control contributions of RISE BSNN (second column) versus time that corresponds to low speed case study 2

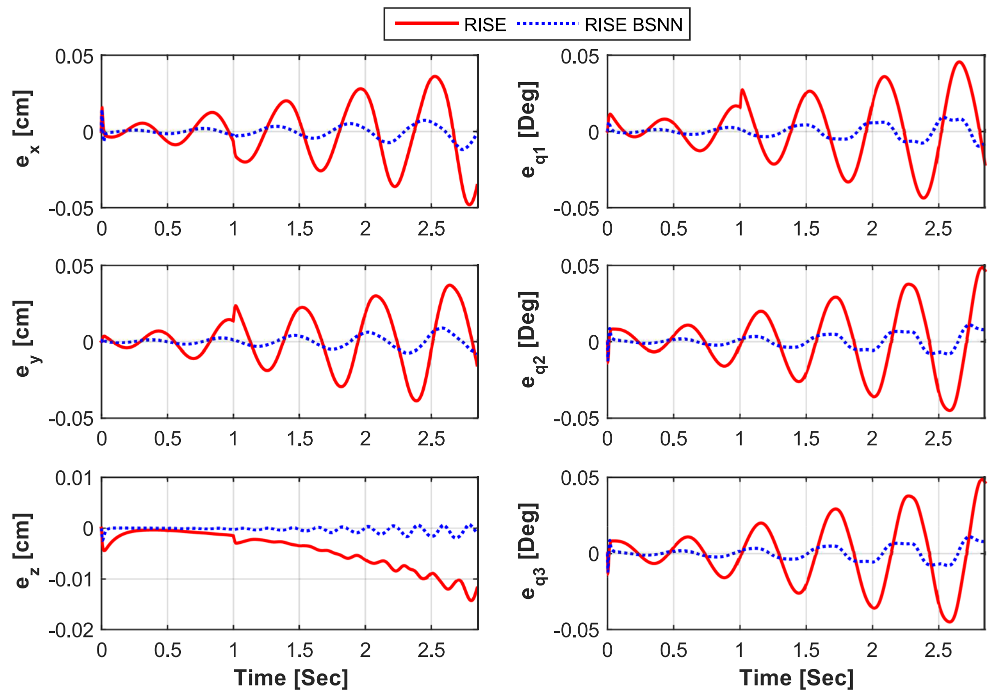


Fig. 15 Evolution of the tracking errors versus time in Cartesian and joint space corresponding to medium speed for case study 2

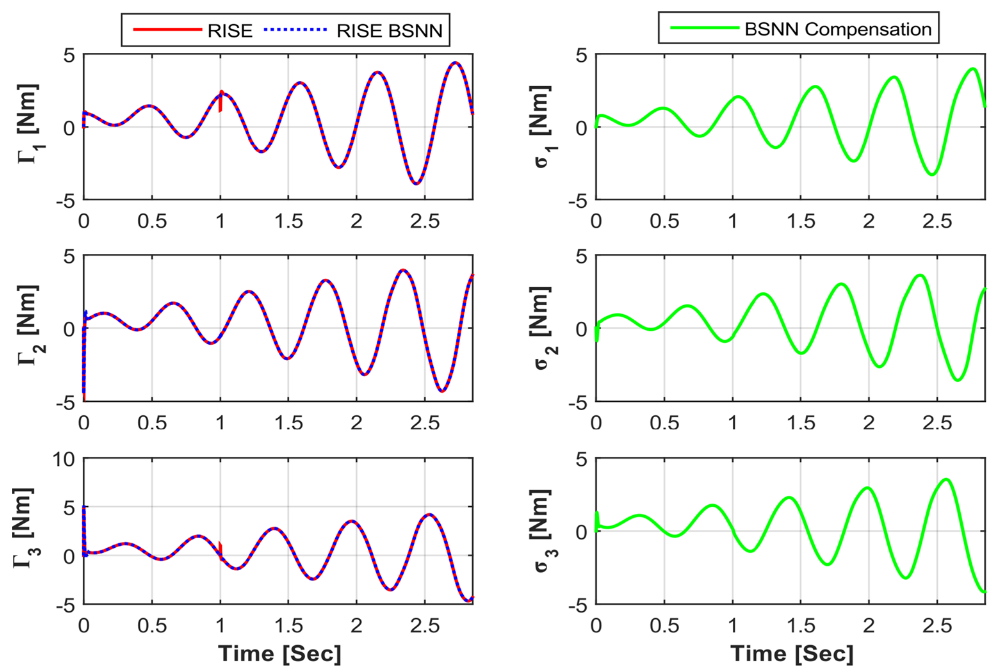
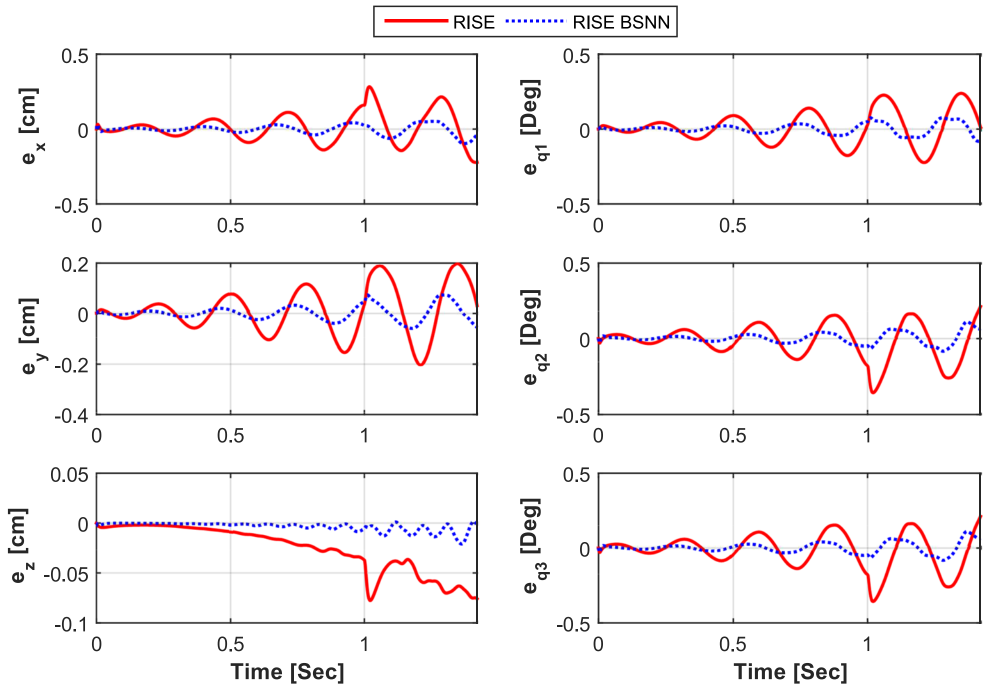


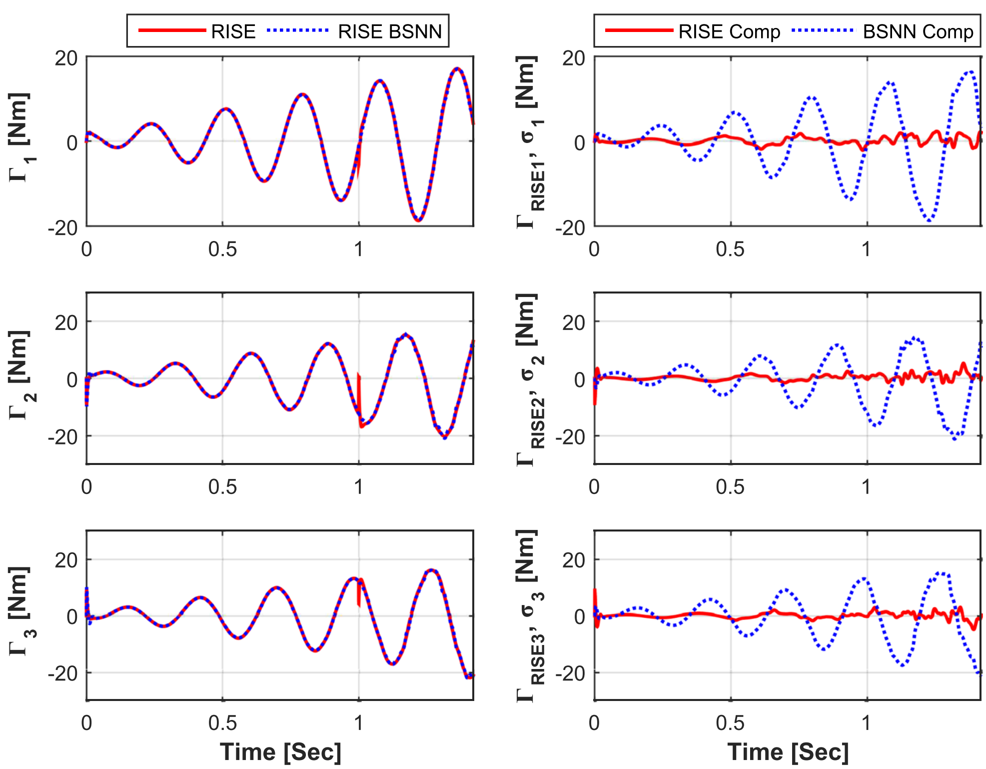
Fig. 16 Evolution of the control signals generated by RISE and RISE BSNN controllers (first column), and the control contributions of RISE BSNN (second column) versus time that corresponds to medium speed case study 2

Fig. 17 Evolution of the tracking errors versus time in Cartesian and joint space corresponding to high speed for case study 2



the results obtained were notably superior. However, as it was mentioned before, the BSNN compensation term aims to emulate the Nominal feedforward term. Therefore, in this section, our proposed control solution is compared to the RISE feedforward, being the combination of Eqs. 27 and 29 to validate the approximation of the dynamics. The case study 1, including the two scenarios, is considered for this validation. Figure 21 depicts the tracking error in the joint space of RISE feedforward and RISE BSNN and the components compensation of both controllers

Fig. 18 Evolution of the control signals generated by RISE and RISE BSNN controllers (first column), and the control contributions of RISE BSNN (second column) versus time that corresponds to medium high case study 2



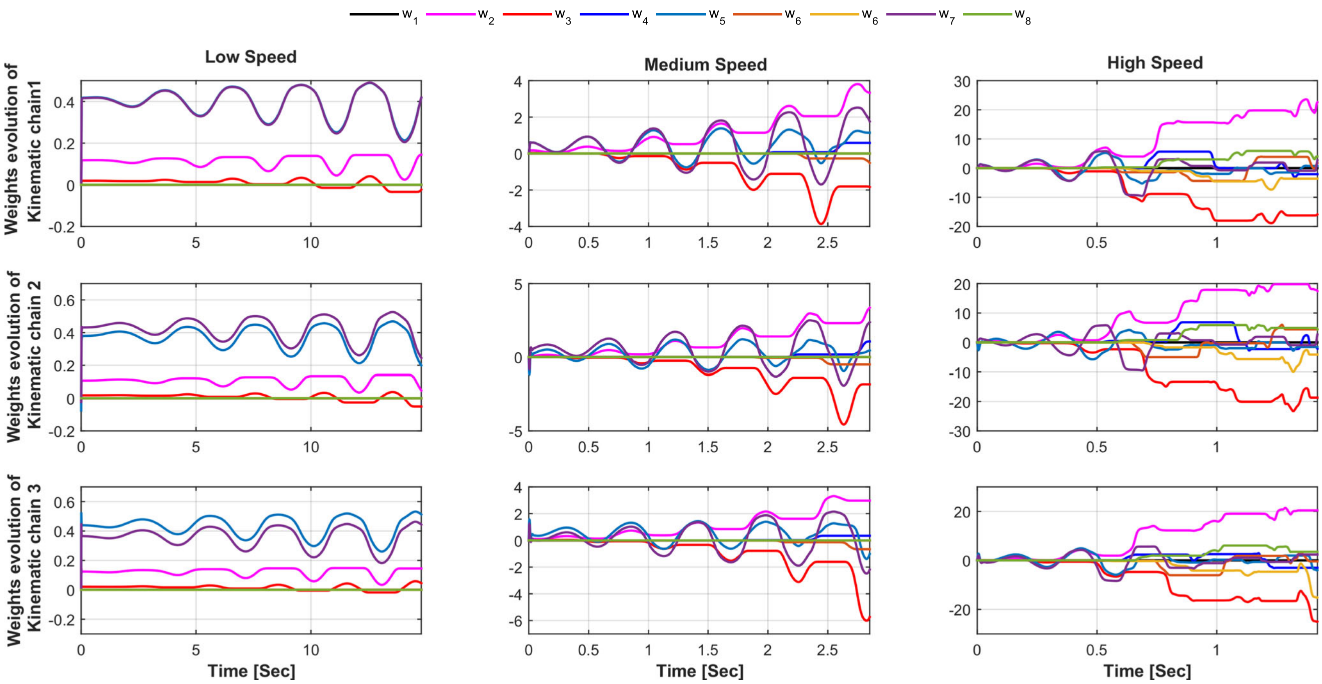
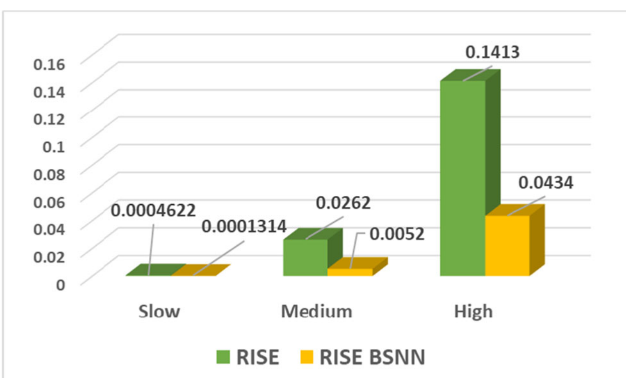
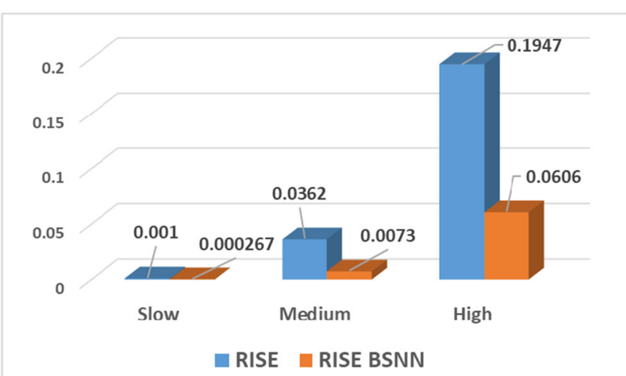


Fig. 19 Evolution of the BSNNs' weights for case study 2 when the DPR is subjected to changes in the speed



(a) RMSE Cartesian Space



(b) RMSE Joint Space

Fig. 20 Degradation graphs of RMSE at different speeds for Cartesian and joint space case study 2

when no payload is moving. It can be appreciated that the tracking error of RISE feedforward is prominently better than our proposition due to the evaluated dynamic parameters in the feedforward part are entirely known, unlike RISE BSNN, where the dynamic behavior of the DPR is on-line estimated. However, note that the produced compensation terms of the BSNN are similar to those produced by the nominal feedforward even without any information on the system dynamics. The obtained $RMSE_q$ is 0.0045 for RISE feedforward and 0.0102 for RISE BSNN, the first controller outcomes the second one in 56.69% for this scenario. Nevertheless, for the second scenario where a mass of 1 kg is moved in some portions of the trajectory, the performance of the RISE BSNN is better than RISE feedforward, due to RISE BSNN can compensate for the parametric uncertainty produced by the changes in the payload along the trajectory, unlike RISE feedforward, where the dynamic parameters are not updated (see Fig. 22). The resulting $RMSE_q$ for the second scenario is 0.0494 for RISE feedforward and 0.0194 for RISE BSNN, yielding an improvement of 60% of RISE BSNN over RISE feedforward.

6 Conclusion

In this work, a RISE controller with adaptive feedforward compensation founded on the BSNN has been proposed.

Table 6 Controllers performance evaluation case study 2

Speed	Controller	**RMSE$_C$[cm]**	**RMSE$_J$[Deg]**
	RISE	4.622×10^{-4}	0.0010
Low	RISE BSNN	1.314×10^{-4}	2.670×10^{-4}
	Enhancement	71.5%	73.6%
	RISE	0.0262	0.0362
Medium	RISE BSNN	0.0052	0.0073
	Enhancement	80.0%	79.8%
	RISE	0.1413	0.1947
High	RISE BSNN	0.0434	0.0606
	Enhancement	69.8%	68.5%

Three BSNN have been implemented in order to approximate the inverse dynamic of each kinematic chain of the DPR. The election of AMNN is mainly due to the low computational cost that carries out this kind of ANN since the computed weights are updated according to the current input value so, not all the weights are updated at the same time. The precise approximation of the inverse dynamics lies mostly in the choosing inputs, the selected order for the basis functions, and the distribution of the knots points. To validate the effectiveness of the proposed control scheme, numerical simulations were performed, the obtained results were compared in a first instance to those of standard RISE controller. The control system was evaluated in two case studies, the first one P&P trajectory execution with changes in the payload, and the second one a spiral path with changes in the speed. For all the scenarios of the case studies, the obtained results showed that the proposed control scheme presented improvements greater than 60%. Thereby, the use of the BSNNs as a feedforward compensation term is a suitable alternative to improving the trajectory tracking in PKMS even if the system is dealing with parametric uncertainties as sudden changes in the payload. Moreover,

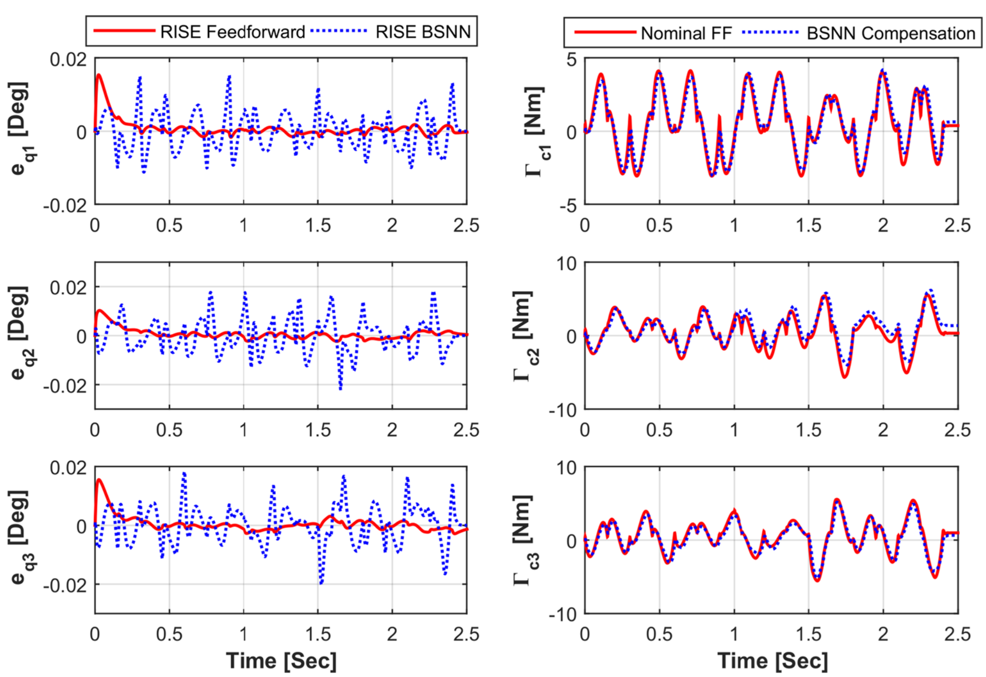


Fig. 21 Performance comparison between RISE feedforward and RISE BSNN scenario 1 case study 1

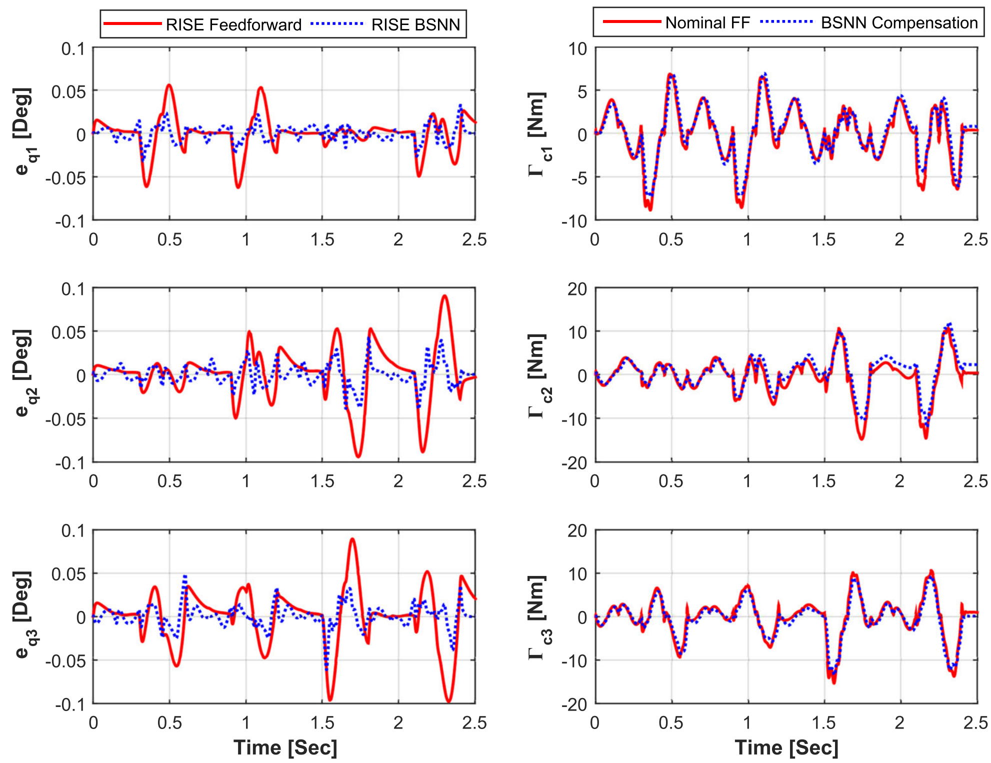


Fig. 22 Performance comparison between RISE feedforward and RISE BSNN scenario 2 case study 1

the dynamic approximation of the BSNNs is good enough according to the comparison of the curves with the nominal Feedforward of a RISE Feedforward controller.

Acknowledgments Thanks to the Program of Support to the development of higher education (PADES), agreement No 2018-13-011-047. Additionally, J. Escorcia thanks to The Mexican Council of Science and Technology (CONACYT); Award no. 593804.

References

1. Briot, S., Khalil, W.: Dynamics of Parallel Robots: From Rigid Bodies to Flexible Elements, vol. 35. Springer, Berlin (2015)
2. Bennehar, M.: Some contributions to nonlinear adaptive control of pkms: from design to real-time experiments. Ph.D. dissertation, Université de Montpellier (2015)
3. Taghirad, H.D.: Parallel Robots: Mechanics and Control. CRC Press, Boca Raton (2013)
4. Brinker, J., Corves, B.: A survey on parallel robots with delta-like architecture. In: Proceedings of the 14th IFToMM World Congress, pp. 407–414 (2015)
5. Luces, M., Mills, J.K., Benhabib, B.: A review of redundant parallel kinematic mechanisms. J. Intell. Robot. Syst. **86**(2), 175–198 (2017)
6. Li, Q., Hervé, J.M., Ye, W.: Geometric Method for Type Synthesis of Parallel Manipulators. Springer, Berlin (2020)
7. Staicu, S.: Dynamics of Parallel Robots. Springer, Berlin (2019)
8. Chemori, A.: Control of complex robotic systems: challenges, design and experiments. In: 2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 622–631. IEEE (2017)
9. Saied, H.: On control of parallel robots for high dynamic performances: From design to experiments. Ph.D. dissertation, Université de Montpellier (2019)
10. Sartori Natal, G.: Control of parallel robots: towards very high accelerations. Ph.D. dissertation, Université Montpellier 2 (2012)
11. Lu, X., Liu, M.: A fuzzy logic controller tuned with pso for delta robot trajectory control. In: Industrial Electronics Society, IECON 2015-41st Annual Conference of the IEEE, pp. 004345–004351. IEEE (2015)
12. Castañeda, L.A., Luviano-Juárez, A., Chairez, I.: Robust trajectory tracking of a delta robot through adaptive active disturbance rejection control. IEEE Trans. Control Syst. Technol. **23**(4), 1387–1398 (2015)
13. Tuvayanond, W., Parnichkun, M.: Position control of a pneumatic surgical robot using pso based 2-dof h∞ loop shaping structured controller. Mechatronics **43**, 40–55 (2017)
14. Rachedi, M., Hemici, B., Bouri, M.: Design of an h∞ controller for the delta robot: experimental results. Adv. Robot. **29**(18), 1165–1181 (2015)
15. Xian, B., Zhang, Y.: A new smooth robust control design for uncertain nonlinear systems with non-vanishing disturbances. Int. J. Control. **89**(6), 1285–1302 (2016)
16. Bennehar, M., Chemori, A., Bouri, M., Jenni, L.F., Pierrot, F.: A new rise-based adaptive control of pkms: design, stability analysis and experiments. Int. J. Control. **91**(3), 593–607 (2018)
17. Saied, H., Chemori, A., Bouri, M., El Rafei, M., Francis, C., Pierrot, F.: A new time-varying feedback RISE control for second-order nonlinear MIMO systems: theory and experiments. International Journal of Control, 1–14. (2019)
18. Escorcia-Hernández, J.M., Chemori, A., Aguilar-Sierra, H., Monroy-Anieva, J.A.: A new solution for machining with ra-pkms: Modelling, control and experiments. Mech. Mach. Theory **150**, 103864 (2020)
19. Zhang, D., Wei, B.: Adaptive Control for Robotic Manipulators. CRC Press, Boca Raton (2017)

20. Bennehar, M., Chemori, A., Pierrot, F.: A novel rise-based adaptive feedforward controller for redundantly actuated parallel manipulators. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2389–2394. IEEE (2014)
21. Tutsoy, O., Erol Barkana, D., Colak, S.: Learning to balance an nao robot using reinforcement learning with symbolic inverse kinematic. Trans. Inst. Meas. Control. **39**(11), 1735–1748 (2017)
22. Tutsoy, O., Barkana, D.E., Tugal, H.: Design of a completely model free adaptive control in the presence of parametric, non-parametric uncertainties and random control signal delay. ISA Trans. **76**, 67–77 (2018)
23. Yu, W.: Pid Control with Intelligent Compensation for Exoskeleton Robots. Academic Press, Cambridge (2018)
24. Deng, H., Srinivasan, D., Oruganti, R.: A b-spline network based neural controller for power electronic applications. Neurocomputing **73**(4), 593–601 (2010)
25. dos Santos Coelho, L., Pessôa, M.W.: Nonlinear identification using a b-spline neural network and chaotic immune approaches. Mech. Syst. Signal Process. **23**(8), 2418–2434 (2009)
26. Asl, H.J., Janabi-Sharifi, F.: Adaptive neural network control of cable-driven parallel robots with input saturation. Eng. Appl. Artif. Intel. **65**, 252–260 (2017)
27. Razmi, M., Macnab, C.J.B.: Near-optimal neural-network robot control with adaptive gravity compensation. Neurocomputing (2020)
28. Asl, H.J., Narikiyo, T., Kawanishi, M.: Neural network-based bounded control of robotic exoskeletons without velocity measurements. Control. Eng. Pract. **80**, 94–104 (2018)
29. Escorcia-Hernández, J.M., Aguilar-Sierra, H., Aguilar-Mejía, O., Chemori, A., Arroyo-Núñez, J.H.: An intelligent compensation through b-spline neural network for a delta parallel robot. In: 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), pp. 361–366 (2019)
30. Corbel, D., Gouttefarde, M., Company, O., Pierrot, F.: Towards 100g with pkm. is actuation redundancy a good solution for pick-and-place? In: 2010 IEEE International Conference on Robotics and Automation, pp. 4675–4682. IEEE (2010)
31. Pierrot, F., Reynaud, C., Fournier, A.: Delta: a simple and efficient parallel robot. Robotica **8**(2), 105–109 (1990)
32. Xian, B., Dawson, D.M., de Queiroz, M.S., Chen, J.: A continuous asymptotic tracking control strategy for uncertain nonlinear systems. IEEE Trans. Autom. Control **49**(7), 1206–1211 (2004)
33. Lopes, J.C., Ruano, A.E., Fleming, P.J.: Identification of aircraft gas-turbine dynamics using b-splines neural networks. IFAC Proc. Volumes **33**(6), 123–128 (2000)
34. Mirea, L.: Dynamic multivariate b-spline neural network design using orthogonal least squares algorithm for non-linear system identification. In: System Theory, Control and Computing (ICSTCC), 2014 18th International Conference, pp. 720–725. IEEE (2014)
35. Brown, M., Harris, C.J.: Neurofuzzy adaptive modelling and control (1994)
36. Khalil, W., Dombre, E.: Modeling, identification and control of robots. Butterworth-Heinemann (2004)
37. Natal, G.S., Chemori, A., Pierrot, F.: Dual-space control of extremely fast parallel manipulators: payload changes and the 100g experiment. IEEE Trans. Control Syst. Technol. **23**(4), 1520–1535 (2015)
38. Atakulreka, A., Sutivong, D.: Avoiding local minima in feedforward neural networks by simultaneous learning. In: Australasian Joint Conference on Artificial Intelligence, pp. 100–109. Springer (2007)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Jonatan Martín Escorcia-Hernández received the B.Sc. degree in Robotic Engineering, and the M.Sc. degree in Automation and Control from the Polytechnic University of Tulancingo, Tulancingo de Bravo, Mexico in 2013 and 2017, respectively. He is currently working towards the Ph.D. in Optomechatronics at the Polytechnic University of Tulancingo. His research interests include modeling and control of robotics systems.

Hipólito Aguilar-Sierra received the B.Sc. degree in Mechatronics Engineering from UPIITA-IPN in 2009; and M.Sc. and Ph. D degrees both in Automatic Control from the CINVESTAV Zacatenco, Mexico City, Mexico, in 2011 and 2016, respectively. He is currently a Full-time professor at Faculty of Engineering from the La Salle Mexico University. His research interests include Medical robots, Rehabilitation robots, Exoskeleton robotics and Nonlinear control.

Omar Aguilar-Mejía received the Engineering degree in Electrical Engineering from Instituto Tecnológico de Pachuca, Hidalgo, México, in 1999; M.Sc. in Electrical Engineering from CINVESTAV Guadalajara, México in 2002; Ph. D. in Industrial Engineering from UAEH, Pachuca, México in 2014. Aguilar is working as Full-time Professor at the postgraduate department at UPAEP University, since 2017. His current research interests are in modeling and control of electromechanical systems and motor drives with computational intelligence techniques.

Ahmed CHEMORI received the M.Sc. and Ph.D. degrees both in automatic control from the Grenoble Institute of Technology, Grenoble, France, in 2001 and 2005, respectively. He has been a Postdoctoral Fellow with the Automatic Control Laboratory, Grenoble, France, in 2006. He is currently a tenured Research Scientist in automatic control and robotics with the Montpellier Laboratory of Informatics, Robotics and Microelectronics (LIRMM-CNRS). His research interests include nonlinear, adaptive and predictive, control and their real-time applications in robotics.

José Humberto Arroyo-Núñez received the M.Sc. degree in 2002 from the National Institute of Astrophysics, Optics and Electronics, Mexico, and the Ph.D. degree in Electronic Engineering from the Universitat Politècnica de València, Spain, in 2016. He is a professor in the Direction of Research and Posgraduate Studies, Polytechnic University of Tulancingo, Mexico. Where he participates in the Doctorate in Optomechatronics and the Master in Automation and Control.

Affiliations

Jonatan Martín Escorcia-Hernández[1] · Hipólito Aguilar-Sierra[2] · Omar Aguilar-Mejia[3] · Ahmed Chemori[4] · José Humberto Arroyo-Núñez[1]

Jonatan Martín Escorcia-Hernández
1715002@upt.edu.mx

Omar Aguilar-Mejia
omar.aguilar@upaep.mx

Ahmed Chemori
ahmed.chemori@lirmm.fr

José Humberto Arroyo-Núñez
humberto.arroyo@upt.edu.mx

1 Universidad Politécnica de Tulancingo, Calle Ingenierías No. 100, C.P. 43629, Tulancingo, Hidalgo, México

2 Facultad de Ingeniería. Universidad La Salle México, Benjamin Franklin No. 45, C.P. 06140, Ciudad de México, México

3 Departamento de Posgrado, UPAEP, Universidad Popular Autónoma del Estado de Puebla, C.P. 72410, Puebla, México

4 CNRS, LIRMM, University of Montpellier, Montpellier, France