

Article

ANN Enhanced Hybrid Force/Position Controller of Robot Manipulators for Fiber Placement

José Francisco Villa-Tiburcio ^{1,*}, José Antonio Estrada-Torres ², Rodrigo Hernández-Alvarado ³,
Josue Rafael Montes-Martínez ³, Darío Bringas-Posadas ¹ and Edgar Adrián Franco-Urquiza ⁴

¹ Center for Engineering and Industrial Development (CIDESI), Manufacturing Department, Av. Playa Pie de la Cuesta No. 702, Queretaro 76125, Mexico; dario.bringas@cidesi.edu.mx

² CONAHCYT-Center for Engineering and Industrial Development (CIDESI), Av. Pie de la Cuesta, No. 702, Queretaro 76125, Mexico; jose.estrada@cidesi.edu.mx

³ Industrial Technologies Division, Polytechnic University of Queretaro (UPQ), Queretaro 76240, Mexico; rodrigo.hernandez@upq.edu.mx (R.H.-A.); josue.montes@upq.edu.mx (J.R.M.-M.)

⁴ Center for Engineering and Industrial Development (CIDESI), Aerospace Division, Carretera Estatal 200, Querétaro-Tequisquiapan KM 23 No. 22547, Localidad Galeras, Colón 76270, Mexico; edgar.franco@cidesi.edu.mx

* Correspondence: jvilla@posgrado.cidesi.edu.mx

Abstract: In practice, most industrial robot manipulators use PID (Proportional + Integral + Derivative) controllers, thanks to their simplicity and adequate performance under certain conditions. Normally, this type of controller has a good performance in tasks where the robot moves freely, performing movements without contact with its environment. However, complications arise in applications such as the AFP (Automated Fiber Placement) process, where a high degree of precision and repeatability is required in the control of parameters such as position and compression force for the production of composite parts. The control of these parameters is a major challenge in terms of quality and productivity of the final product, mainly due to the complex geometry of the part and the type of tooling with which the AFP system is equipped. In the last decades, several control system approaches have been proposed in the literature, such as classical, adaptive or sliding mode control theory based methodologies. Nevertheless, such strategies present difficulties to change their dynamics since their design consider only some set of disturbances. This article presents a novel intelligent type control algorithm based on back-propagation neural networks (BP-NNs) combined with classical PID/PI control schemes for force/position control in manipulator robots. The PID/PI controllers are responsible for the main control action, while the BP-NNs contributes with its ability to estimate and compensate online the dynamic variations of the AFP process. It is proven that the proposed control achieves both, stability in the Lyapunov sense for the desired interaction force between the end-effector and the environment, and position trajectory tracking for the robot tip in Cartesian space. The performance and efficiency of the proposed control is evaluated by numerical simulations in MATLAB-Simulink environment, obtaining as results that the errors for the desired force and the tracking of complex trajectories are reduced to a range below 5% in root mean square error (RMSE).

Keywords: automated fiber placement; complex surface; neural network; classical control; force/position tracking controller; Lyapunov stability



Citation: Villa-Tiburcio, J.F.; Estrada-Torres, J.A.; Hernández-Alvarado, R.; Montes-Martínez, J.R.; Bringas-Posadas, D.; Franco-Urquiza, E.A. ANN Enhanced Hybrid Force/Position Controller of Robot Manipulators for Fiber Placement. *Robotics* **2024**, *13*, 105. <https://doi.org/10.3390/robotics13070105>

Academic Editor: Kensuke Harada

Received: 21 May 2024

Revised: 24 June 2024

Accepted: 26 June 2024

Published: 13 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The products required in sectors such as aerospace, submarines, automotive, wind energy, and naval are becoming increasingly demanding in terms of material type, mechanical properties, and customized geometry. To meet these demands, it has been necessary to test numerous technological developments in the manufacturing area to improve efficiency and quality. One such innovation is the Automated Fiber Placement (AFP) manufacturing

process [1,2]. AFP is one of the most advanced, popular, and established automated composite manufacturing techniques used to produce structures with highly complex geometry, including surfaces with different radii and high degrees of curvature. AFP has attracted significant interest as an advanced composite manufacturing technology due to its high efficiency, repeatability, and effective reduction of labor intensity [3–5]. The automation of this manufacturing process is achieved through the use of a placement head (AFP head) mounted on the end effector of a manipulator robot. This head positions, heats, and consolidates the resin-impregnated fiber ribbons in a manner that is highly similar to manual lamination [6,7]. To achieve optimal results in the final product, it is essential to precisely follow the desired trajectories of the complex surface and maintain the compaction force at the desired value. This requires the AFP head to be accurately positioned and oriented by the manipulator robot [8–11]. Figure 1 illustrates the AFP system on a complex surface mold, showcasing the main components of the system (own design).

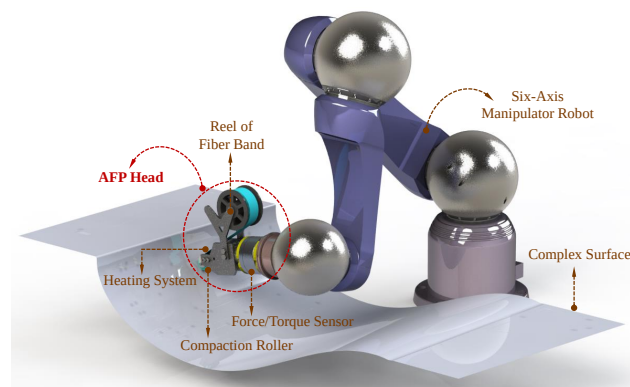


Figure 1. 3D model of the six-axis robot with an automated fiber placement head.

The AFP manufacturing system presents a significant challenge due to its non-linear and multivariable nature. It is susceptible to a range of external disturbances, including interactions with geometry, the addition or removal of new components due to wear, the integration of new technological developments, and electrical noise. These disturbances can have a significant impact on the system's performance. The technological capabilities of the AFP system have been gradually enhanced by advances in robotics and mechatronics over the past few decades. However, printing fiber placement remains a challenge due to the high complexity of surface tracking and compaction force control. This presents an important field of study and development for both the academic and industrial communities [12,13].

In the AFP process, it is essential that the robot tool maintains contact with the surface. Therefore, it is necessary to moderate the position of the tool and the force exerted to ensure optimal performance. Furthermore, it is essential to maintain the tool at a constant angle relative to the surface in order to ensure uniform pressure at all contact points. There are numerous approaches to this type of control problem, such as hybrid force and position control of robotic manipulators, which can be found in current literature. In [14], a position and force control scheme for automated fiber placement was presented. The proposed method is integrated by a force control scheme based on exteroceptive sensor loops to control the compaction force and a visual control scheme given by a camera to control the lateral position of the tape. The objective is to improve the accuracy of fiber placement and, subsequently, the quality of the fabricated parts. In [15], the authors proposed a variable stiffness admittance control strategy (NVTs) for force and position tracking of a manipulator to perform polishing applications on flat surfaces. This strategy employs a variable stiffness proportional-integral-derivative (PID) law, which is designed to update the stiffness coefficient of the admittance function in response to feedback from the force and position. In [16], a novel constant position and force tracking control scheme is proposed, based on an in-line stiffness and inverse damping force stiffness impedance controller

(OSRDF). The objective is to track the desired force in robotic finishing operations. The proposed approach tracks the desired contact force and reference trajectory as a function of the reference position and velocity. In a similar vein, [17] presents an impedance control for curved surface tracking and contact force regulation of a manipulator robot for robotic tooling applications such as polishing and grinding. The surface geometry is tracked and approximated through sensors installed on the robot end-effector. In [18], a coupled force-position control (CFPC) method for a manipulator robot is presented. This method combines the proposed adaptive impedance control method with a modified HFPC method, which is designed for use in situations where the surface is inclined, curved, or otherwise complicated. In [19], a novel symmetric internal and external adaptive variable impedance symmetric control for the position/force tracking of welding process-oriented multi-robot cooperative manipulators is proposed. In [20], the authors present a study of the complex surface polishing process, wherein the gravity-compensated admittance control method is employed to adjust the position and force of the manipulator robot tool. In [21], a hybrid approach to force/position control of a robot manipulator for the machining of complex parts (aero-engine blade) was proposed. The proposed controller incorporates force calculation, zero drift compensation, and gravity compensation, and is combined with proportional-integral/proportional-derivative (PI/PD) controllers to maintain a desired position and contact force between the manipulator and the complex environment. In [22], a robust hybrid position/force control (HPFC) system for multi-degree-of-freedom (MDoF) manipulators with torque constraints at each joint is proposed. The proposed method simultaneously handles the response characteristics and torque constraints of the actuators by using predictive functional control (PFC) and the model parameter errors are compensated by the disturbance observer technique. Additionally, studies have been conducted with the objective of addressing general industrial requirements. For instance, the work [23] presents a novel symmetrical adaptive variable admittance control methodology for the position/force tracking of two-arm cooperative manipulators. In a similar vein, a novel voltage-based weighted hybrid force/position control algorithm for redundant robotic manipulators is proposed in [24] with the objective of guaranteeing position and force accuracies during contact operations. In addition, contributions have been made in other fields, such as space [25], where the force/position tracking control problem of a free-flying space manipulator with uncertain kinematics and dynamics interacting with an uncertain docile surface is solved through the design of an adaptive Jacobian controller. Another field in which this approach has been applied is medicine. In [26], the authors put forward a comprehensive adaptive admittance control strategy for the position and force control of an ultrasound scanning robotic system operating in a soft and uncertain environment. While these controls have become a viable option for position and force control, their results are limited due to the presence of unknown dynamics, such as friction, model uncertainties and external disturbances. These factors can significantly impact the accuracy of these controllers.

Other control techniques that have been successfully employed for the position/force control of manipulator robots include those based on adaptive sliding mode control (ASMC) theory. This control strategy is well-suited for the cleaning of handrails, escalators and other flat surfaces [27], as well as for surface treatment such as polishing, grinding, finishing and deburring [28]. Other authors have addressed this problem by employing control techniques such as continuous terminal sliding mode control (CTSMA) [29], super-twist sliding mode control (STSMC) [30], or model-independent adaptive fractional high-order terminal sliding mode control (AFO-HoTSMC) [31]. The main challenge with these control techniques is the number of design parameters involved, which makes it difficult to identify the optimal values for achieving optimal dynamic performance.

The deployment of observers has also been instrumental in enhancing the reliability of hybrid force/position control systems. To illustrate, in [32–34], observers are employed to quantify disparate types of disturbances, including unknown loads, unmodeled dynamics, and environmental interaction forces. In other documented studies, they are utilized as

estimators of joint velocities [35], contact forces [36,37], or external interaction torques [38]. However, the construction of an observer faces significant challenges, including unreliable sensor signals, system vibrations, and in most cases, it is necessary to perform a preliminary identification process of the plant or to obtain the parameters of the observer that guarantee an adequate convergence.

In addition to the aforementioned control schemes, some academics have also employed more sophisticated control techniques, such as intelligent algorithms based on neural networks, due to their ability to detect and compensate for disturbances in systems where high precision is mandatory. In [39], we propose a force/position tracking impedance control scheme based on radial basis function neural networks (RBFNN) and adaptive Jacobian to control a 2 DoF robotic system with external uncertainties and disturbances. In [40], an adaptive hybrid impedance control (AHIC) scheme based on radial basis function neural networks (RBFNN) is proposed as a means of approximating saturated errors, uncertain parts and external disturbances in flexibly articulating robotic manipulators. In [41], an adaptive neural control strategy for a 2 DoF manipulator with unknown non-linearities and parametric uncertainties is proposed. The strategy employs a Radial Basis Function Neural Network (RBFNN) to address the unknown dynamics, thereby ensuring that the tracking error and estimation error exponentially converge to a small set around zero. In [42], an adaptive impedance neural control (NAIC) scheme with backpropagation was proposed for the position and desired force tracking of a 7-degree-of-freedom (DoF) robot that is in contact with an unknown environment. In [43], a backpropagation neural network (BP-NN) is employed to track a reference force signal and compensate for the influence of unknown parameters associated with a two-degree-of-freedom (2 DoF) manipulator and the environment. In [44], feedforward neural networks with sliding windows are used to control the position of a 1 DoF robot, the neural network weight coefficients are updated online using the actuator position data as the controller is applied to the system. Another type of neural network with excellent computational power and hardware feasibility is the recurrent neural network (RNN). This has been employed as a new form of noise rejection and position-force control in 6 and 7 degree-of-freedom manipulator robots [45–47]. The efficacy of neural networks has been evidenced by their low computational cost and reduced error rate, which have facilitated their deployment in predictive control systems [48] and observers [49]. Other researchers have even explored the potential of combining these intelligent algorithms with other control schemes, this has led to the development of new combinations as PID-Neuronal (NN-PID) [50], Neuro-Sliding (NN-SMC) [51–53] and Neuro-Fuzzy (NN-FL) [54,55].

The results presented in the above studies were effective in simulation and practical study. However, in most of them, the controller was implemented for the first 3 DoFs on manipulator robots having at least 6 DoFs. As a consequence, most of the results are oriented to non-complex surfaces, which does not provide a picture that guarantees the efficiency and robustness of the proposed method for surfaces with higher degree of complexity, as is the case of the AFP process. In addition, it has been observed that some of these methods involve long computational processes due to the high number of operations, which makes them unsuitable for practical application.

This article presents a straightforward and effective force/position control strategy for n -degree-of-freedom (DoF) manipulator robots oriented towards the automated fibre placement (AFP) manufacturing process. The general control strategy comprises backpropagation artificial neural networks (BP-NNs), which are integrated with classical proportional-integral-derivative/proportional-integral (PID/PI) control schemes. The proposed strategy employs two NNs with an identical topological structure, one for the control of the position loop and the other for the control of the force loop, thereby generating the control structure known as hybrid force/position control. In comparison with the work cited, the contributions of this study are summarized as follows:

1. The combination of a neural network with classical proportional-integral-derivative (PID) and proportional-integral (PI) controllers compensates for the inherent simplicity

of these types of controllers with the introduction of artificial intelligence, thereby enhancing their resilience to significant alterations in their parameters. The primary objective of NNs is to offset the dynamic effects that arise when the manipulator interacts with its surrounding environment.

2. Another important point is that the weight coefficients of the NN are updated on-line, without prior training, using the force/position data from the sensors and actuators, the output of the conventional controller and the errors generated in the force/position tracking.
3. The proposed strategy does not require a large amount of computational resources, its structure is simple and it can be implemented on a real platform without any data collection or training process for any n DoF manipulator robot.

The rest of this article is organized as follows. The dynamic model of the robot and some useful properties are presented as a preliminaries in Section 2. The development of the intelligent control algorithm and the stability analysis are presented in Section 3. The methodology for the generation of complex test trajectories is presented in Section 4. The 3D virtual simulation scenario and the discussion of the results are presented in Section 5. Finally, the conclusions are presented in Section 6.

2. Preliminaries

Dynamic Model and Robot Properties

The dynamic model of an n DoF manipulator robot can be described as:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D(\dot{q}) + g(q) + \tau_p = \tau - \tau_{ext} \quad (1)$$

where q , $\dot{q}$, and $\ddot{q} \in \mathbb{R}^n$ denote the joint position, velocity, and acceleration, respectively; $M(q) \in \mathbb{R}^{n \times n}$ is the positive definite symmetric inertia matrix; $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the matrix of centripetal-Coriolis forces; $D \in \mathbb{R}^n$ is a positive semidefinite diagonal matrix representing viscous friction; $g \in \mathbb{R}^n$ is the vector of gravitational pairs; $\tau_p \in \mathbb{R}^n$ is the vector of perturbations; $\tau \in \mathbb{R}^n$ is the vector of control pairs; and $\tau_{ext} \in \mathbb{R}^n$ is the vector of external torques resulting from the external force, $F_{ext} \in \mathbb{R}^n$, exerted by the manipulator on the environment at the contact point

$$\tau_{ext} = J^T(q)F_{ext} \quad (2)$$

where $J^T(q) \in \mathbb{R}^{n \times n}$ represents the transpose of the Jacobian matrix [42]. The robot dynamics given in Equation (1) has the following useful properties [56].

Property 1. The inertia matrix $M(q) \in \mathbb{R}^{n \times n}$ is symmetric and positive definite, and also the inverse matrix $M(q)^{-1}$ exists and is positive definite:

$$M(q) > 0, M^{-1}(q) > 0 \implies x^T M(x)x > 0 \quad \forall x \in \mathbb{R}^n \quad (3)$$

Property 2. The centripetal-Coriolis matrix $C(q, \dot{q})$ and the derivative with respect to time of the inertia matrix $\dot{M}(q)$ satisfy the relationship:

$$\dot{q}^T \left[\frac{1}{2} \dot{M}(q) - C(q, \dot{q}) \right] \dot{q} = 0 \quad \forall q, \dot{q} \in \mathbb{R}^n \quad (4)$$

where the matrix $[\dot{M}(q) - 2C(q, \dot{q})]$ is an antisymmetric matrix.

Property 3. From Property 2 the derivative with respect to time of the inertia matrix and the matrix centripetal-Coriolis matrix satisfy:

$$\dot{M}(q) = C(q, \dot{q}) + C(q, \dot{q})^T \quad (5)$$

Property 4. The left side of the Equation (1) is linear with respect to the selected set of robot and load parameters, i.e.,

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D(\dot{q}) + g(q) = Y\Theta \quad (6)$$

where $Y = Y(q, \dot{q}, \ddot{q}) \in \mathbb{R}^{n \times p}$ $Y \Theta \in \mathbb{R}^p$ contains the unknown parameters of the manipulator robot and the load.

Assumption 1. The unknown perturbations are bounded by:

$$\|\tau_p\| \leq \tau_N \quad (7)$$

where τ_N is a known positive constant.

3. Adaptive Neural Network for Force/Position Control in Manipulator Robots

To ensure homogeneous fiber placement during the AFP process, the orientation of the robot end-effector must be perpendicular to the mold surface, i.e., the end-effector must be held in an orthogonal position to the plane tangent to the surface at each contact point, as shown in Figure 2; another important parameter to consider is the applied tool force, which must be constant throughout the stroke [2].

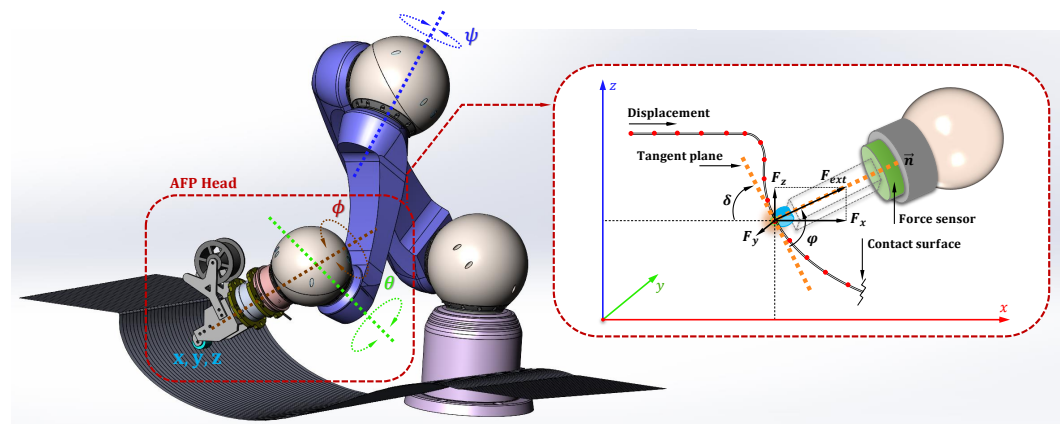


Figure 2. Tool-workpiece contact state during AFP process.

In summary, for AFP process the deviation angle φ and the contact force F_{ext} , must satisfy the following conditions:

$$\begin{cases} \varphi = \frac{\pi}{2} \\ F_{ext} = F_d \end{cases} \quad (8)$$

this means that it is necessary that the normal vector, $\vec{n}$, to the tangent plane at the AFP head contact point with the surface is parallel to that of the contact force F_{ext} and the magnitude of the latter regulated at the desired value F_d [57,58]. In order to control F_{ext} , the magnitude of such a force is measured by the force sensor integrated at the AFP head (see Figure 2).

The force sensor measures the applied stress in the six degrees of freedom, this provides an output data vector $F_x, F_y, F_z, M_x, M_y, M_z$ as shown in Figure 3.

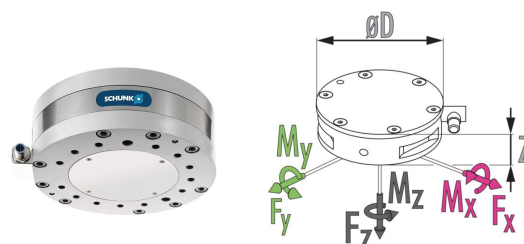


Figure 3. Rigid 6-axis force/torque sensor [59].

identity matrix; $\tau_q = \tau_{pid} + \tau_{nnq}$, $\tau_f = \tau_{pi} + \tau_{nnf} \in \mathbb{R}^n$ are the composite control signals and are computed separately. The control action τ_{pid} is described by:

$$\tau_{pid} = K_{pp}\tilde{q} + K_{ip} \int_0^t \tilde{q} dt + K_{dp}\dot{\tilde{q}} \quad (13)$$

where $\tilde{q}$ represents the position tracking error; $\dot{\tilde{q}}$ corresponds to the velocity error; K_{pp} , K_{ip} , K_{dp} are positive definite proportional, integral and derivative gain matrices, respectively [62]. On the other hand, the control action τ_{pi} , described by:

$$\tau_{pi} = K_{pf}\tilde{f} + K_{if} \int_0^t \tilde{f} dt \quad (14)$$

where $\tilde{f}$ represents the force tracking error; K_{pf} and K_{if} are positive definite proportional and integral gain matrices for the force [62].

The NN representing the control actions τ_{nnq} and τ_{nnf} is mathematically represented by the Equation (15), where $\sigma(\cdot)$ is the activation function, w_{ij} is the input to hidden layer interconnection weights, and v_{jk} is the hidden to output layer interconnection weights.

$$\tau_{nn} = \sigma \left[\sum_{j=1}^1 v_{jk} \sigma \left[\sum_{i=1}^n w_{ij} x_i + w_i \right] + v_j \right] \quad (15)$$

The Equation (15) of the NN can be rewritten in matrix form as:

$$h = V^T \sigma(W^T x) + \epsilon \quad (16)$$

where $h = \tau_{nn}$, $x = [x_1, x_2, x_3, \dots, x_i]^T$, $W^T = [w_{ij}]^T$ is the optimal weight value of the hidden layer, $V^T = [v_{jk}]^T$ is the optimal weight value of the output layer, ϵ is an approximation error vector NN bounded by $\|\epsilon\| \leq \epsilon_N$ for some $\epsilon_N > 0$ [63]. Replacing both control actions in Equation (1), the dynamic model of the manipulator robot would be expressed:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D(\dot{q}) + g(q) + \tau_p = \tau_\eta - \tau_{ext} \quad (17)$$

where τ_η is the proposed control law and is represented by the following equation:

$$\tau_\eta = K_{pp}\tilde{q} + K_{ip} \int_0^t \tilde{q} dt + K_{dp}\dot{\tilde{q}} + K_{pf}\tilde{f} + K_{if} \int_0^t \tilde{f} dt + V^T \sigma(W^T x) + \epsilon \quad (18)$$

3.2. ANN for Position Control

The topological structure of the NN used for position control is shown in Figure 5a. The NN consists of nine neurons in the input layer, three neurons in the hidden layer and one neuron in the output layer, where $q_d(t)$ and $q_d(t - t_s)$ are the desired position at current time and desired position one sampling time before, similarly, $q_r(t)$ and $q_r(t - t_s)$ correspond to actual position outputs, $\tau_{pid}(t)$ and $\tau_{pid}(t - t_s)$ are the classical PID control signals, $\tau_q(t)$ and $\tau_q(t - t_s)$ are the PID+NN associated control signals, $\tilde{q}(t)$ is the path tracking error; finally, τ_{nnq} corresponds to the compensation signal of the NN to achieve the desired position. The sigmoid function was used for the activation of the neurons of the hidden layer w_{ij} as well as for the neuron of the output layer v_{jk} , this real function with domain: $\mathbb{R} \rightarrow (-\infty, +\infty)$ and range: $\mathbb{R} \rightarrow (0, 1)$ has the advantage of being differentiable infinitely many times, a fundamental aspect when the learning process is carried out by means of the back-propagation algorithm. The back-propagation consists of moving in the negative direction of the gradient of the cost function $E(x, y)$ with respect to the coefficients w_{ij} and v_{jk} , as shown in Figure 6 [64].

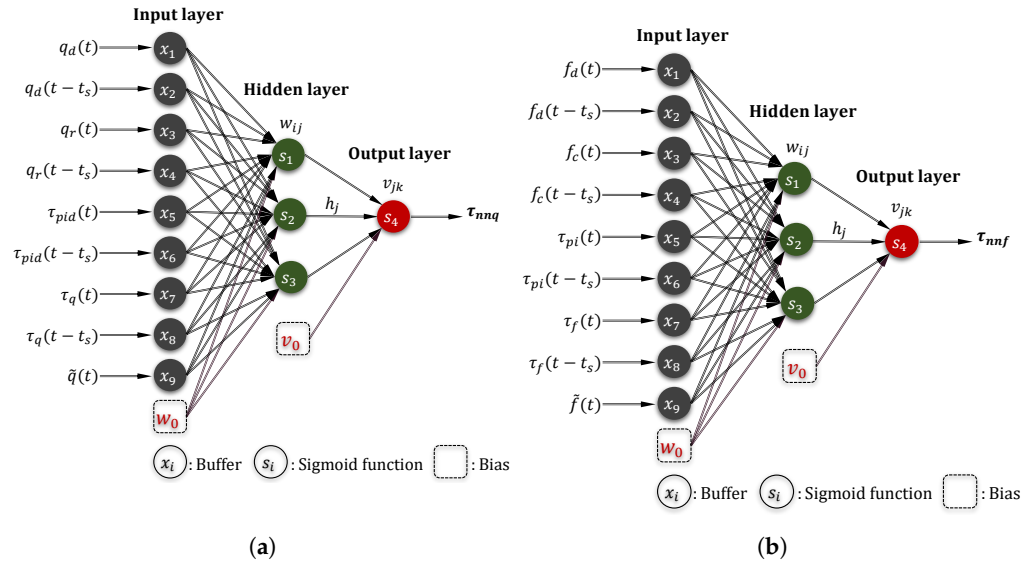


Figure 5. NN topology for (a) position control, and (b) force control.

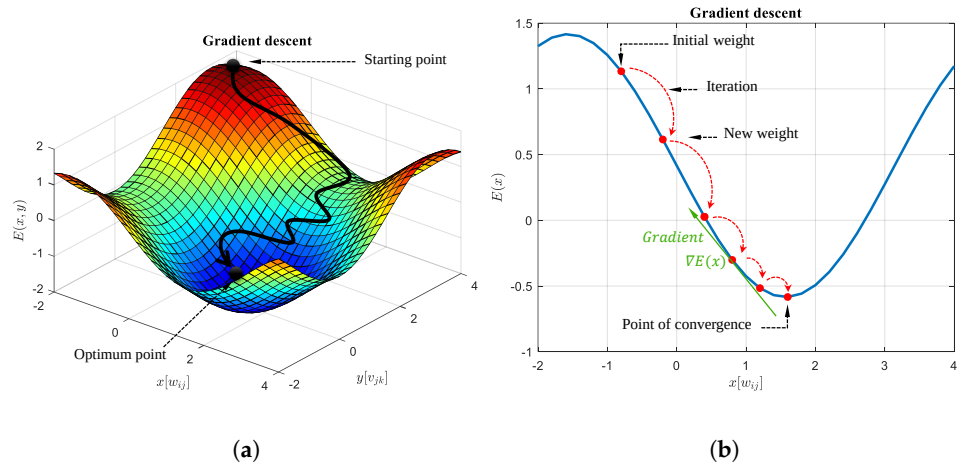


Figure 6. NN cost function in (a) 3D, and (b) 2D.

Developing the intelligent algorithm starts by computing the output of the hidden layer neurons by the following function:

$$h_j = \frac{1}{1 + e^{-s_j}} \quad (19)$$

where

$$s_j = \sum_{i=1}^3 w_{ij}x_i + w_i \quad (20)$$

Similarly, the value of the output layer, scaled by a gain factor of λ_1 , is determined by:

$$\tau_{nnq} = \frac{\lambda_1}{1 + e^{-r_k}} \quad (21)$$

where

$$r_k = \sum_{j=1}^1 v_{jk}h_j + v_j \quad (22)$$

The criterion to be minimized is defined as the following cost function:

$$E_q(t) = \frac{1}{2} \sum_{k=1}^t (\tilde{q})^2 \quad (23)$$

where $\tilde{q}$ is the error generated in the trajectory tracking given by:

$$\tilde{q} = q_r - q_d \quad (24)$$

The gradient of the position $E_q(t)$ is a multidimensional vector whose components are the partial derivatives $\frac{\partial E_q(t)}{\partial v_{jk}}$ and $\frac{\partial E_q(t)}{\partial w_{ij}}$, i.e.,

$$\nabla E_q(t) = \begin{bmatrix} \frac{\partial E_q(t)}{\partial v_{jk}} \\ \frac{\partial E_q(t)}{\partial w_{ij}} \end{bmatrix} \quad (25)$$

The following partial derivatives express the first component of the gradient $E_q(t)$:

$$\begin{aligned} \frac{\partial E_q(t)}{\partial v_{jk}} &= \frac{\partial E_q(t)}{\partial \tilde{q}} \frac{\partial \tilde{q}}{\partial e_{\tau_q}} \frac{\partial e_{\tau_q}}{\partial \tau_{nnq}} \frac{\partial \tau_{nnq}}{\partial r_k} \frac{\partial r_k}{\partial v_{jk}} \\ &= \tilde{q} \frac{\partial \tilde{q}}{\partial e_{\tau_q}} (-1) \lambda_1 \tau_q (1 - \tau_q) \end{aligned} \quad (26)$$

The following relation was used for the term $\frac{\partial \tau_{nnq}}{\partial r_k}$:

$$\begin{aligned} \frac{\partial \tau_{nnq}}{\partial r_k} &= \frac{\partial \left(\frac{\lambda_1}{1 + e^{-r_k}} \right)}{\partial r_k} \\ &= \lambda_1 \tau_q (1 - \tau_q) \end{aligned} \quad (27)$$

By defining:

$$\delta_k^1 = \tilde{q} \lambda_1 \tau_q (1 - \tau_q) \quad (28)$$

Then $\frac{\partial E_q(t)}{\partial v_{jk}}$ is expressed:

$$\frac{\partial E_q(t)}{\partial v_{jk}} = -\delta_k^1 h_j \frac{\partial \tilde{q}}{\partial e_{\tau_q}} \quad (29)$$

The second component of the gradient, $E_q(t)$, is the result of the application of the chain rule again:

$$\begin{aligned} \frac{\partial E_q(t)}{\partial w_{jk}} &= \frac{\partial E_q(t)}{\partial \tilde{q}} \frac{\partial \tilde{q}}{\partial q_r} \frac{\partial q_r}{\partial \tau_{nnq}} \frac{\partial \tau_{nnq}}{\partial r_k} \frac{\partial r_k}{\partial h_j} \frac{\partial h_j}{\partial s_j} \frac{\partial s_j}{\partial w_{ij}} \\ &= -\tilde{q} \frac{\partial \tilde{q}}{\partial e_{\tau_q}} \lambda \tau_q (1 - \tau_q) v_{jk} h_j (1 - h_j) x_i \\ &= -\delta_k^1 v_{jk} h_j (1 - h_j) x_i \frac{\partial \tilde{q}}{\partial e_{\tau_q}} \end{aligned} \quad (30)$$

By defining:

$$\delta_j^2 = \delta_k^1 v_{jk} h_j (1 - h_j) \quad (31)$$

Then $\frac{\partial E_q(t)}{\partial w_{ij}}$ is expressed:

$$\frac{\partial E_q(t)}{\partial w_{ij}} = -\delta_j^2 x_i \frac{\partial \tilde{q}}{\partial e_{\tau_q}} \quad (32)$$

Using the Equations (29) and (32), the coefficients v_{jk} and w_{ij} are adjusted using the expressions:

$$v_{jk} = v_{jk} + \left(\alpha \frac{\partial \tilde{q}}{\partial e_{\tau_q}} \right) \delta_k^1 h_j \quad (33)$$

$$w_{ij} = w_{ij} + \left(\alpha \frac{\partial \tilde{q}}{\partial e_{\tau_q}} \right) \delta_j^2 x_i \quad (34)$$

where α is the learning constant used for position and $\frac{\partial \tilde{q}}{\partial e_{\tau_q}}$ can be interpreted as the equivalent gain of the unknown [65].

3.3. ANN for Force Control

For force control, we used the topological structure of the NN shown in Figure 5b, where $f_d(t)$ and $f_d(t - t_s)$ correspond to the desired force input; $f_r(t)$ and $f_r(t - t_s)$ is the actual force output provided by the force sensor (see Figure 3); $\tau_f(t)$ and $\tau_f(t - t_s)$ is the signal from the classical PI controller; $\tilde{f}(t)$ and $\tilde{f}(t - t_s)$ is the control signal assigned by the PI+NN; $\tilde{f}(t)$ is the error of the contact force, and finally the τ_{nnf} corresponds to the compensation signal of the NN to achieve the desired force. The cost function, as a criterion to be minimized, is defined by:

$$E_f(t) = \frac{1}{2} \sum_{k=1}^t (\tilde{f})^2 \quad (35)$$

where $\tilde{f}$ is the force tracking error given by:

$$\tilde{f} = f_r - f_d \quad (36)$$

The gradient of the force $E_f(t)$ is defined by:

$$\nabla E_f(t) = \begin{bmatrix} \frac{\partial E_f(t)}{\partial v_{jk}} \\ \frac{\partial E_f(t)}{\partial w_{ij}} \end{bmatrix} \quad (37)$$

The partial derivatives of the two components are expressed:

$$\frac{\partial E_f(t)}{\partial v_{jk}} = \frac{\partial E_f(t)}{\partial \tilde{f}} \frac{\partial \tilde{f}}{\partial e_{\tau_f}} \frac{\partial e_{\tau_f}}{\partial \tau_{nnf}} \frac{\partial \tau_{nnf}}{\partial r_k} \frac{\partial r_k}{\partial v_{jk}} \quad (38)$$

$$\frac{\partial E_f(t)}{\partial w_{jk}} = \frac{\partial E_f(t)}{\partial \tilde{f}} \frac{\partial \tilde{f}}{\partial f_r} \frac{\partial f_r}{\partial \tau_{nnf}} \frac{\partial \tau_{nnf}}{\partial r_k} \frac{\partial r_k}{\partial h_j} \frac{\partial h_j}{\partial s_j} \frac{\partial s_j}{\partial w_{ij}} \quad (39)$$

Solving both terms with respect to the weights coefficients of the output layer v_{ij} and hidden layer w_{ij} , we obtain:

$$\frac{\partial E_f(t)}{\partial v_{jk}} = -\delta_k^3 h_j \frac{\partial \tilde{f}}{\partial e_{\tau_f}} \quad (40)$$

$$\frac{\partial E_f(t)}{\partial w_{ij}} = -\delta_j^4 x_i \frac{\partial \tilde{f}}{\partial e_{\tau_f}} \quad (41)$$

Define δ_k^3 and δ_j^4 as follows:

$$\delta_k^3 = \tilde{f} \lambda_2 \tau_f (1 - \tau_f) \quad (42)$$

$$\delta_j^4 = \delta_k^3 v_{jk} h_j (1 - h_j) \quad (43)$$

The coefficients v_{ij} and w_{ij} for the force are adjusted in accordance with the following equations:

$$v_{jk} = v_{jk} + \left(\beta \frac{\partial \tilde{f}}{\partial e_{\tau_f}} \right) \delta_k^3 h_j \quad (44)$$

$$w_{ij} = w_{ij} + \left(\beta \frac{\partial \tilde{f}}{\partial e_{\tau_f}} \right) \delta_j^4 x_i \quad (45)$$

where β is the learning constant used for the force and $\frac{\partial \tilde{f}}{\partial e_{\tau_f}}$ can be interpreted as the equivalent gain of the unknown plant [65].

3.4. Stability Analysis

According to Property 4, Equation (6) can be written in terms of a nominal reference $\dot{q}_r$ and its derivative $\ddot{q}_r$:

$$M(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + D(\dot{q}) + g(q) = \hat{Y}\Theta \quad (46)$$

where $\hat{Y} = \hat{Y}(q, \dot{q}, \ddot{q}_r, \ddot{q}_r) \in \mathbb{R}^{n \times p}$.

Subtracting Equation (46) from Equation (17) gives the open-loop error equation:

$$M\dot{s} = \tau_\eta - \tau_{ext} - \tau_p - \hat{Y}\Theta - Cs \quad (47)$$

The approximate value of the output $\hat{Y}\Theta$ is given by the following expression:

$$\hat{h}(x) = \hat{Y}\Theta \equiv \hat{V}^T \sigma(\hat{W}^T x) \quad (48)$$

where $\hat{V}^T, \hat{W}^T$ are the current values of the NN weights obtained by the weight adjustment algorithm specified in the previous section.

Substituting the proposed control action given by Equations (18) and (48) into Equation (47) gives the closed-loop error dynamics:

$$M\dot{s} = -K_{Dp}s - K_{Df}r + \tilde{h} + \tau_p - Cs \quad (49)$$

where $s = \dot{q} - \dot{q}_r$ and $r = \tilde{f} - \dot{f}_r$ is the extended error in position and force, the terms $K_{Dp}s$ and $K_{Df}r$ are known as damping injections, these terms act as a mechanical brake or damper; $K_{Dp} = K_{Dp}^T \in \mathbb{R}^{n \times n}$, $K_{Df} = K_{Df}^T \in \mathbb{R}^{n \times n}$ are positive definite gain matrices. On the other hand, $\tilde{h} = h - \hat{h}$, or put another way:

$$\begin{aligned} \tilde{h} &= V^T \sigma(W^T x) - \hat{V}^T \sigma(\hat{W}^T x) + \epsilon(x) \\ &= \tilde{V}^T \sigma(\tilde{W}^T x) + \epsilon(x) \end{aligned} \quad (50)$$

where $\tilde{V} = V - \hat{V}$, $\tilde{W} = W - \hat{W}$. Define $\hat{\sigma}^T = \frac{d\hat{\sigma}(\hat{V}^T x)}{d\hat{V}^T x}$, for the scalar sigmoid case $\sigma'(z) = \sigma(z)(1 - \sigma(z))$.

Theorem 1. Let the dynamics of the n -DOF manipulator given by Equation (1) be controlled by τ_η , given by Equation (18), then the closed-loop system given by Equation (49) is such that the tracking errors with respect to force, $\tilde{f}$, position $\tilde{q}$, and joint velocity, $\tilde{\dot{q}}$ are asymptotically stable provided that the control gains are positive and the weights w_{ij} and v_{jk} are adapted according to:

$$\dot{\hat{W}} = K_W \sigma(x) s^T \quad (51)$$

$$\dot{\hat{V}} = K_V x (\hat{\sigma}^T \hat{W})^T. \quad (52)$$

Proof. According to Lyapunov stability analysis, if the Lyapunov function is positive definite and its derivative is negative semidefinite, then the control system is stable. Therefore, we choose the Lyapunov function as follows to ensure the stability of the control system:

$$V = \frac{1}{2}s^T Ms + \frac{1}{2}tr\{\tilde{W}^T K_W^{-1} \tilde{W}\} + \frac{1}{2}tr\{\tilde{V}^T K_V^{-1} \tilde{V}\} \quad (53)$$

which includes quadratic functions of the error and quadratic functions of the learning weights in order to study stability of the whole system. Differentiating, we get:

$$\begin{aligned} \dot{V} &= \frac{1}{2}\dot{s}^T Ms + \frac{1}{2}s^T \dot{M}s + \frac{1}{2}\dot{s}^T Ms + \frac{1}{2}tr\{\tilde{W}^T K_W^{-1} \dot{\tilde{W}}\} + \frac{1}{2}tr\{\tilde{V}^T K_V^{-1} \dot{\tilde{V}}\} \\ &= s^T \dot{M}s + \frac{1}{2}s^T \dot{M}s + \frac{1}{2}tr\{\tilde{W}^T K_W^{-1} \dot{\tilde{W}}\} + \frac{1}{2}tr\{\tilde{V}^T K_V^{-1} \dot{\tilde{V}}\} \end{aligned} \quad (54)$$

so that substituting the Equation (49) we get

$$\begin{aligned} \dot{V} &= s^T(-K_{Dp}s - K_{Df}r + \tilde{h} + \epsilon + \tau_p - Cs) + \frac{1}{2}s^T \dot{M}s + \frac{1}{2}tr\{\tilde{W}^T K_W^{-1} \dot{\tilde{W}}\} + \frac{1}{2}tr\{\tilde{V}^T K_V^{-1} \dot{\tilde{V}}\} \\ &= s^T(-K_{Dp}s - K_{Df}r + \tilde{h} + E) + \frac{1}{2}s^T(\dot{M}s - 2C)s + \frac{1}{2}tr\{\tilde{W}^T K_W^{-1} \dot{\tilde{W}}\} + \frac{1}{2}tr\{\tilde{V}^T K_V^{-1} \dot{\tilde{V}}\} \end{aligned} \quad (55)$$

where $E = (\epsilon + \tau_p)$, and using the Property 2, we get

$$\dot{V} = -s^T K_{Dp}s - s^T K_{Df}r + s^T \tilde{V}^T \sigma(\tilde{W}^T x) + s^T E + \frac{1}{2}tr\{\tilde{W}^T K_W^{-1} \dot{\tilde{W}}\} + \frac{1}{2}tr\{\tilde{V}^T K_V^{-1} \dot{\tilde{V}}\} \quad (56)$$

Using the law of adaptive learning given by the Equations (51) and (52), we obtain:

$$\dot{V} = -s^T K_{Dp}s - r^T K_{Df}r + s^T(\epsilon + \tau_p) \quad (57)$$

Since ϵ is bounded above by ϵ_N and by Assumption 1, the Equation (57) is expressed as follows:

$$\dot{V} = -s^T K_{Dp}s - r^T K_{Df}r \quad (58)$$

Since $V > 0$ and $\dot{V} \leq 0$, this shows stability in the Lyapunov sense, so that s , r , $\tilde{V}$ and $\tilde{W}$ (and hence $\hat{V}$, $\hat{W}$) are bounded. Finally, we conclude that the origin of the closed-loop system of Equation (49) is locally asymptotically stable, which implies that the force error $\tilde{f} \rightarrow 0$, the position error $\tilde{q} \rightarrow 0$ and the velocity error $\dot{\tilde{q}} \rightarrow 0$. $\square$

4. Complex Trajectory Generation

4.1. Surface for Trajectory Tracking Test

To perform the simulation tests of the AFP process, the surface shape of a wind turbine blade section was designed as a solid part using SolidWorks software. The complexity of the surface is highlighted by the following views: front view, top view, and right side view, as shown in Figure 7.

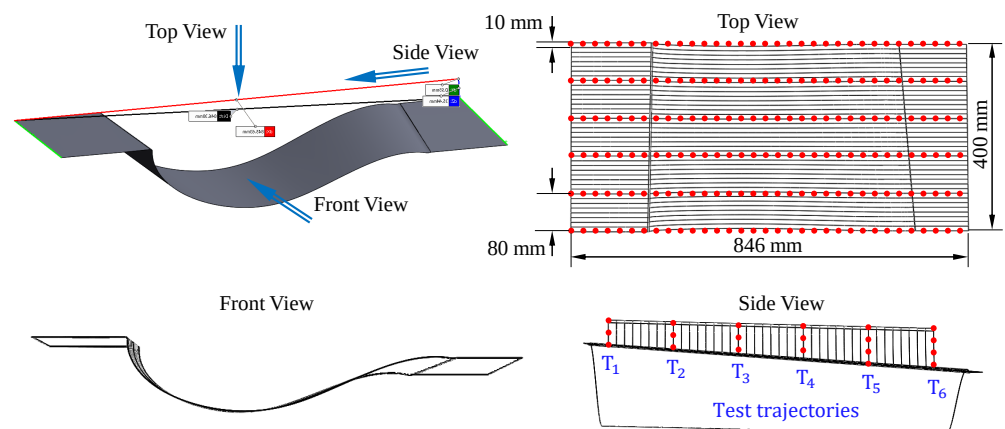


Figure 7. Turbine blade section used for simulation force/position control tests.

Due to the dimensions of the model, only one section was taken for the simulations (see Figure 7). The section taken was divided into strips evenly distributed along the length and width of the surface, with a spacing of 10 mm according to the width of the fiber to be used for an intended application [2]. Forty-one trajectories were generated, marked with a black solid line, of which six were selected for the simulation. These trajectories are the desired cartesian coordinates (x_d, y_d, z_d) depicted on the block diagram in Figure 2.

4.2. Methodology Overview

The process of reconstructing a surface from a point cloud is a complex task that depends on the accuracy of the algorithm to guarantee the approximation of the final surface to the original points as well as on the quality required for the surface [15]. In the Figure 8 is shown in schematic form the proposed methodology for the generation of the desired trajectories, developed in MATLAB software, the objective is to calculate a surface as close as possible to the section of the selected CAD (Computer-Aided Design) model.

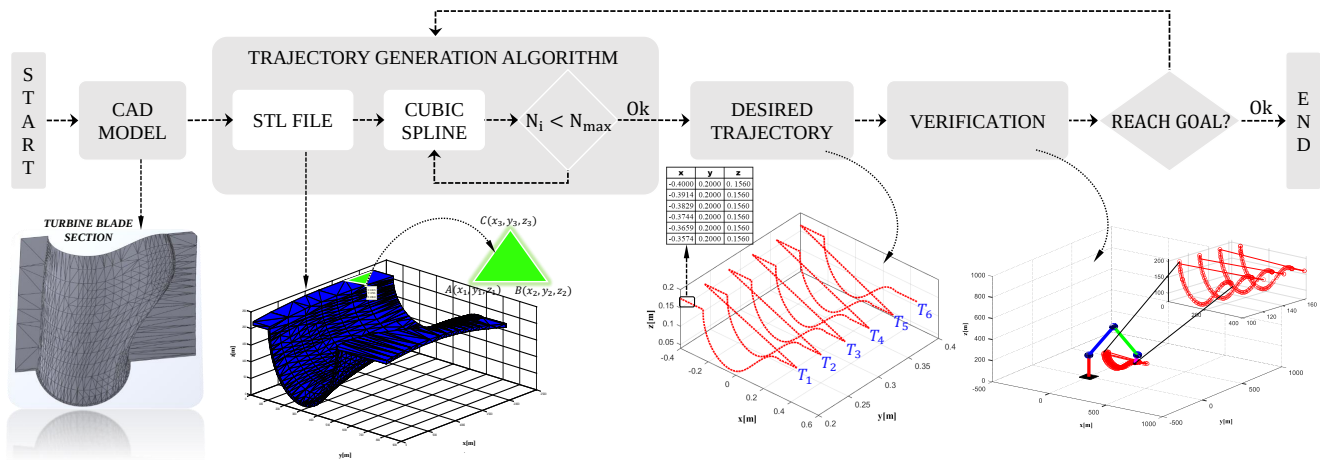


Figure 8. Proposed methodology for the generation and validation of complex trajectories.

The following steps describe the process:

1. The methodology starts with the CAD design of the surface, the CAD model consists of a section of a wind turbine blade whose main characteristic is the complexity of its geometry.
2. We continue to export the CAD model as an STL (Standard Triangulation Language) file, these files are a 3D CAD computer file format that defines the geometry of 3D objects in the form of triangles, excluding information such as color, textures or physical properties included in other 3D CAD formats.
3. STL files are processed using the trajectory generation algorithm, which includes the following tasks:
 - Generate a cloud of points with coordinates (x, y, z) , these points correspond to the corners of reading triangles that must be debugged in a way that the manipulator robot can interpret to perform the AFP manufacturing process.
 - The generated points are grouped and sorted with a separation of 10 units forming trajectories. The separation will depend on the width of the fiber to be used in the process.
 - Debug the generated trajectories by eliminating overlapping points.
 - Apply the cubic spline interpolator with the following restriction: the number of points generated N_i by trajectory must be less than the maximum number of points N_{max} that the manipulator robot can process, i.e., $N_i < N_{max}$.
4. After applying the algorithm, the new trajectories obtained are saved in a file with extension ".txt" with coordinates (x, y, z) .

5. The process continues with verification, this is done by tracking tests with a 6-axis robot manipulator. The comparison with the original part is done by calculating the RMSE, this is presented in Section 5.

The pseudocode is presented in Appendix A.

5. Results Discussion

It is a formidable challenge to model all aspects involved in position/force control of robots dealing with surface treatment tasks. Several phenomena affect the system, such as sensor noise, different sensor behavior in free motion and contact motion, communication delays, material stiffness, etc. In summary, it is a highly uncertain non-linear system. In this section we test the performance and robustness of the proposed approach only in simulation scenarios, leaving as future work its validation on a real platform. The proposed approach was subjected to the following simulation scenarios:

- (a) SCENARIO I. In this scenario, the effectiveness of the intelligent control algorithm is tested under ideal conditions by following the test trajectories generated in Section 4.
- (b) SCENARIO II. In this scenario, the robustness of the intelligent control algorithm is tested in the presence of variable disturbances, such as the effects of tool friction on the contact surface and unmodeled dynamics. Subsequently, a comparison is made with the classical PID/PI controllers, both controls evaluated under the same conditions.

5.1. Intelligent Control Execution for Complex Surface Tracking with Force Control

The tests performed in each scenario can be considered as a two-phase movement: the first one is a free motion, that is, when the robot moves freely towards the environment, and the second phase starts once the robot is in contact with the complex environment. We consider the 6 DoF Schunk Powerball LWA 4P manipulator robot model that is shown in Figure 1. Its dynamic parameters are included in [66]. The proposed approach was evaluated in a 3D virtual simulation environment developed in MATLAB/Simulink, using the ODE 45 integration method with an integration step of $t = 0.001$ s. The computer control platform of the robot Schunk Powerball LWA 4P is an Intel(R) Core(TM) i7-1065G7 CPU with 1.30 GHz processor and 16.0 GB RAM under Windows 11.

SCENARIO I. For this case no perturbations are considered, it is aimed to evaluate the PID/PI controllers with adjustable neural compensation by tracking the complex trajectories generated in the previous section: $T_1, T_2, T_3, T_4, T_5, T_6$. The initial conditions for the position and orientation of the end-effector are as follows: $x = -0.4$ m, $y = 0.2$ m, $z = 0.32$ m, $\psi = 0$, $\theta = \frac{\pi}{2}$, $\phi = 0$. The manipulator robot begins with an initial error of 0.004357 m in position. A constant feed rate of $v = 0.01$ m/s was set for the tool. The magnitude of the contact force was set to a desired value of $f_d = 10$ N. The values of the gains for the PID control, obtained heuristically, were the following: $K_{pp} = \text{diag}([70, 70, 40, 40, 40, 40])$, $K_{ip} = \text{diag}([0.5, 0.5, 0.5, 0.5, 0.5, 0.5])$ y $K_{dp} = \text{diag}([30, 30, 30, 30, 30, 30])$ while the gains for the PI control, were the following: $K_{pf} = \text{diag}([90, 90, 60, 60, 60, 60])$, $K_{if} = \text{diag}([0.2, 0.2, 0.2, 0.2, 0.2, 0.2])$. The initial values for the weight coefficients for the neurons in the hidden layer, w_{ij} , and the output layer, v_{jk} , were set to -1 , 0 , and 1 . The learning coefficients, α and β , were assigned a value of 0.5 for each degree of freedom of the manipulator robot.

Figure 9 presents a 3D dynamic visualization of the Schunk Powerball LWA 4P manipulator tracking six trajectories in a 3D virtual environment.

For a more detailed analysis of the system performance, please refer to Figures 10–15, which have been enlarged in some parts up to 95%.

The results of the numerical simulation of tracking the six trajectories in a satisfactory manner with isometric, top, front, and right side views can be seen in more detail in Figure 10. The evaluation of the trajectories was conducted in accordance with the directions of the arrows marked in orange. The green solid line represents the actual tool trajectory (xyz_r), while the red dashed line represents the desired trajectory (xyz_d).

Figure 11 illustrates the movements and errors generated by the tool along the x, y, z axes. On the left side, it can be seen that the settling time is nearly identical for each axis,

achieved in less than 0.1 s with minimal overshoot while the weights (w_{ij}, v_{jk}) are adapted. On the right side, it can be seen that the errors at steady state are minimal, thus maintaining contact and correct surface tracking.

Figure 12 illustrates the evolution of the robot links over time in response to control torques, as well as the errors generated during the path. The controller's robustness is reflected in the low joint and Cartesian errors.

Figure 13 illustrates the magnitude of the linear velocity of the tool, maintained at a desired value of 0.01 m/s.

The left side of the Figure 14 shows the value of the tool's deflection angle φ . It is evident that maintaining a good orthogonal orientation of the tool with respect to the complex surface is possible, with the value remaining around $\frac{\pi}{2}$ throughout the trajectory. In contrast, the force generated by the interaction between the tool and the environment is represented in the right part of the same Figure 14. While a slight overshoot is evident at the start of the trajectory, the control system is able to maintain the desired force level of 10 N during the interaction. These numerical results demonstrate that the conditions stated in Equation (8) are satisfied.

It is also crucial to monitor the behavior of the control torques to prevent damage to the manipulator motors. Figure 15 illustrates the control torques for each of the joints. Small start duration peaks can be observed on τ_2, τ_3, τ_5 . This is due to the effort required to transition from free motion to the restricted one when the manipulator contacts the surface plane.

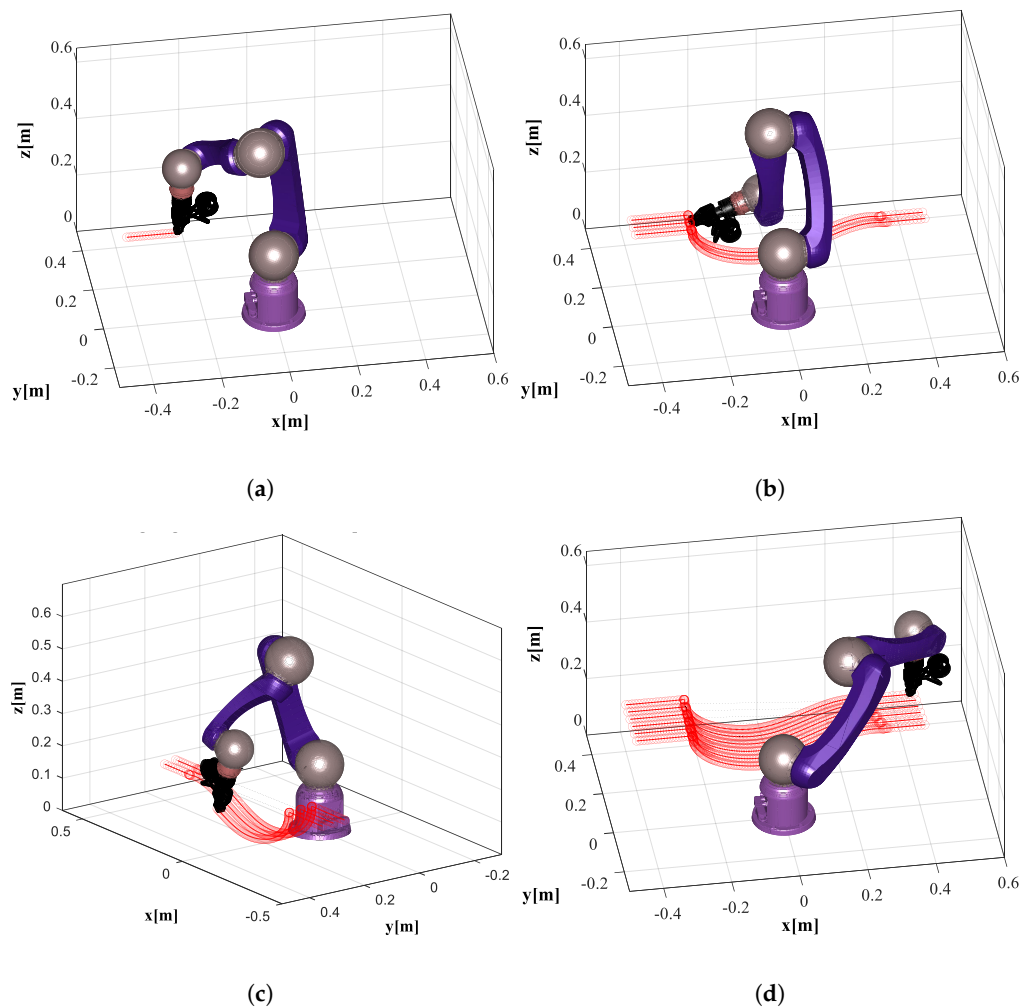


Figure 9. Snapshots of the robot manipulator during the trajectory tracking along a complex surface for (a) $t = 20$ s, (b) $t = 648$ s, (c) $t = 775$ s, and (d) $t = 1700$ s.

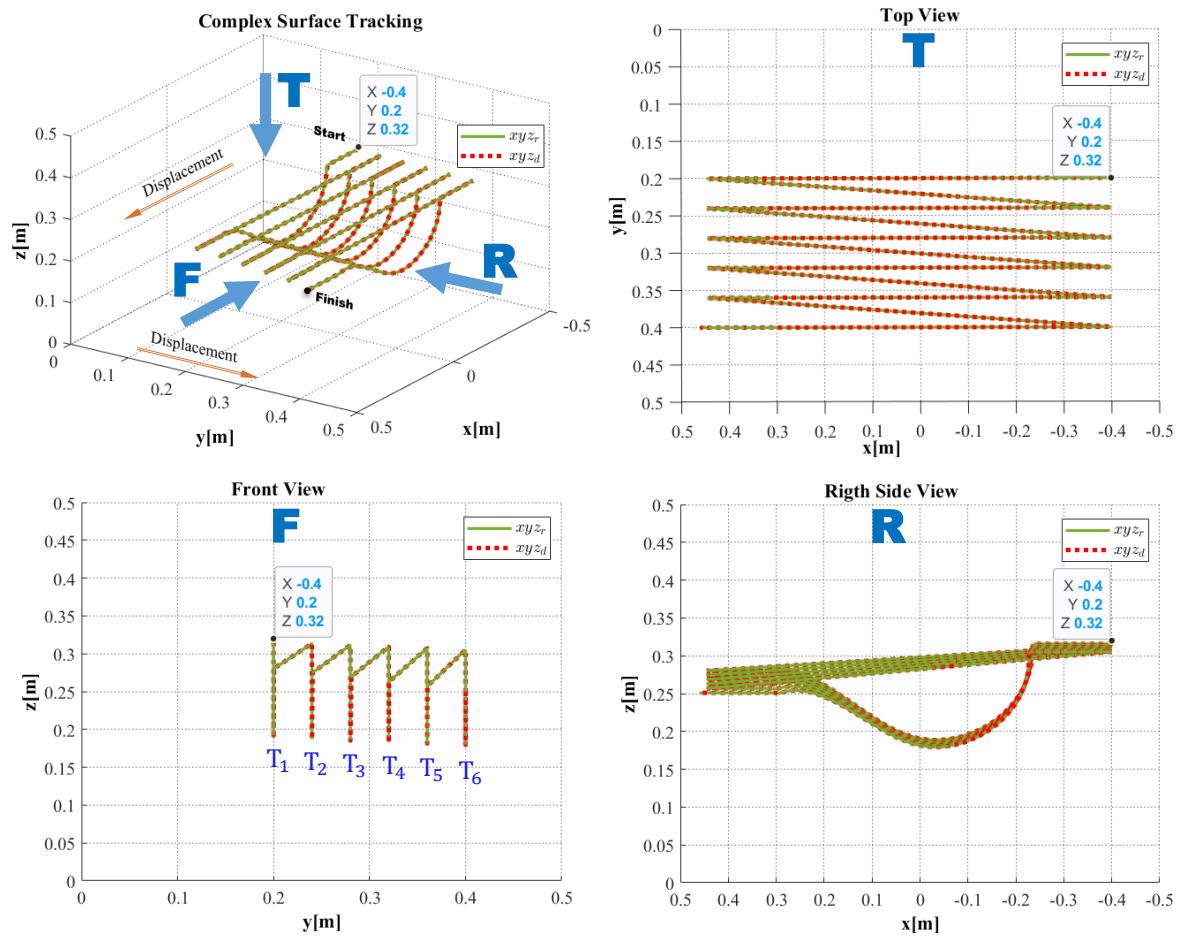


Figure 10. Complex trajectory tracking in Cartesian space: real trajectory xyz_r (green solid line) vs. desired trajectory xyz_d (red dotted line).

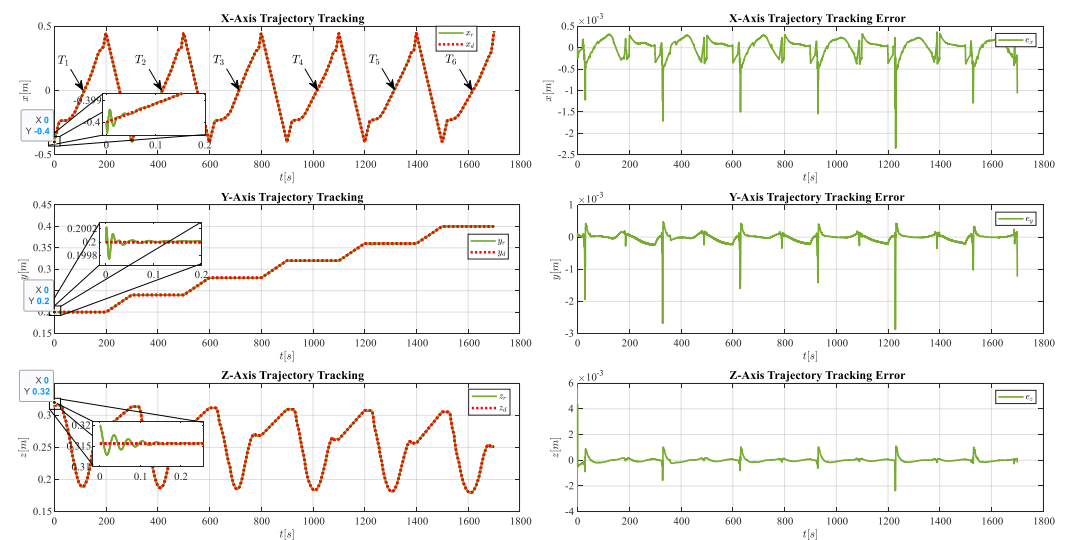


Figure 11. Position tracking (left) y error position tracking (right) on the axes x, y, z .

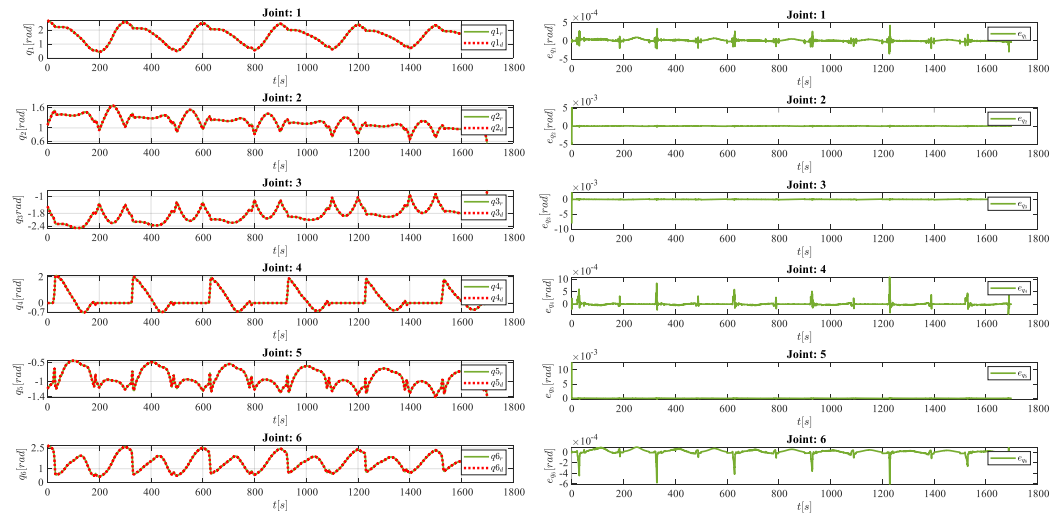


Figure 12. Movement of the six joints and angular (left) angular errors (right) of the manipulator robot.

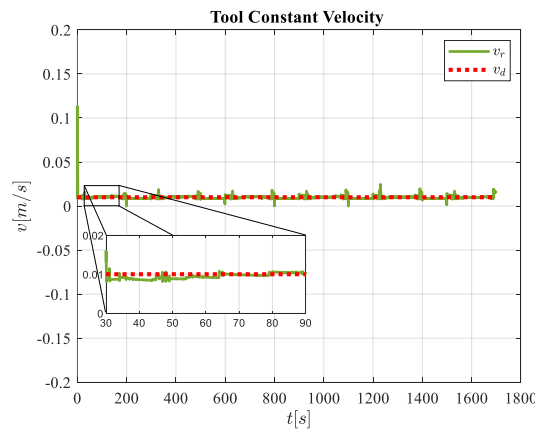


Figure 13. Real linear velocity v_r (green solid line) vs. desired v_d (red dotted line).

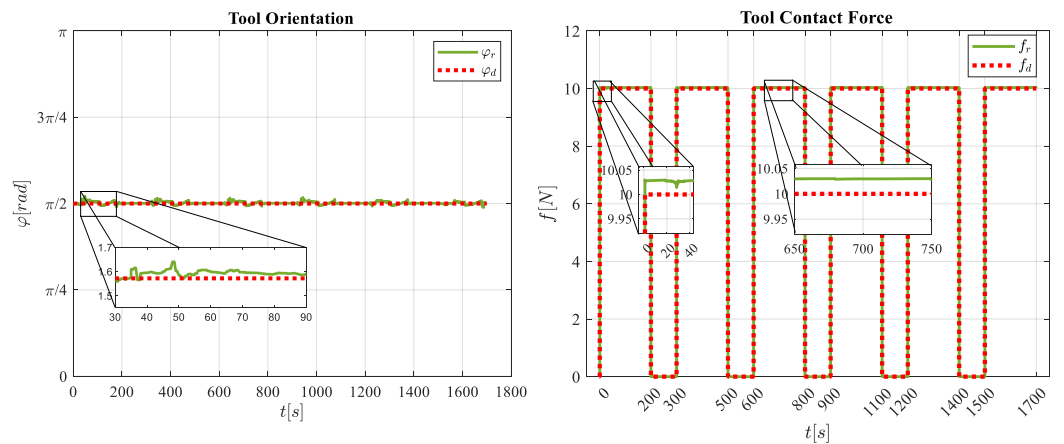


Figure 14. Numerical results of orientation (left) and force (right) along the complex surface. Actual orientation φ_r (green solid line) desired φ_d (red dotted line). Real compaction force f_r (green solid line) vs. desired f_d (red dotted line).

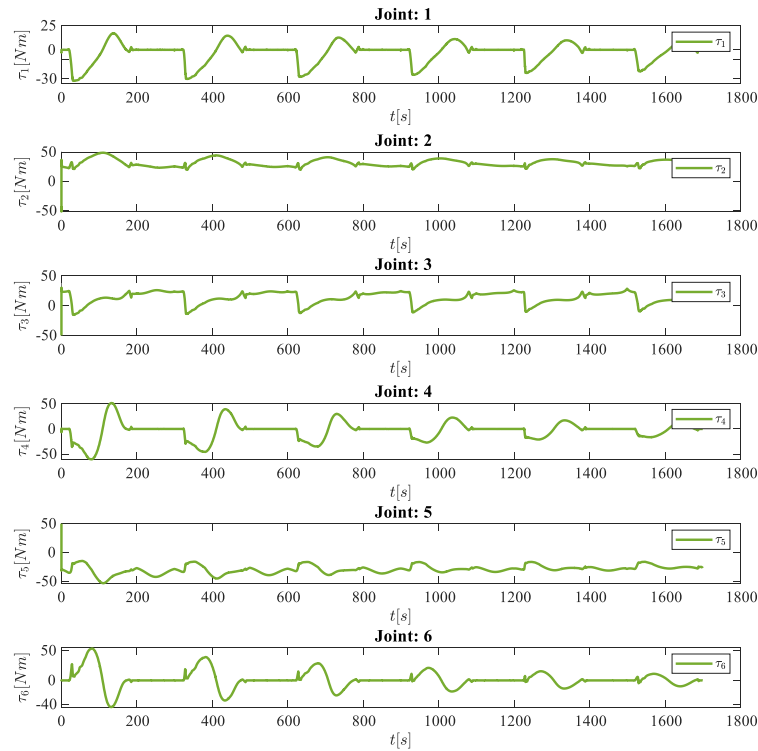


Figure 15. Control torques values.

5.2. Controllers Comparison: Classical PID vs. Adaptive PID-NN

SCENARIO II. This scenario involves three tests based on the trajectory tracking T_1 . In this scenario, we compare the proposed control algorithm with the classical PID/PI controllers developed for trajectory tracking and force control. We maintain the values of the gains of the PID/PI controllers, as well as the initial conditions for w_{ij} and v_{jk} and their learning coefficients α and β . New initial conditions are set for the position and orientation of the end-effector: $x = -0.4$ m, $y = 0.1$ m, $z = 0.45$ m, $\psi = 0$, $\theta = \frac{\pi}{2}$, $\phi = 0$. The initial error in position with which the manipulator robot starts is increased to a value of 0.167486 m, which is challenging given the characteristics of the system. Test 1 considers trajectory tracking under an ideal environment (without perturbation). Tests 2 and 3 present the presence of varying external perturbations. The simulation time was $t = 200$ s.

Test 1. Figures 16–18 show the results of the manipulator robot's behavior in position and force under ideal conditions.

The results presented in Figure 16 demonstrate that the reference tracking is adequate and significantly improved for the PID+NN control in comparison to the classical PID control.

Figure 17 illustrates that the error in the PID-NN trajectory tracking rapidly approaches zero and remains at that value for the duration of the 200 s analysis.

Figure 18 illustrates the force generated by the interaction between the manipulator tool and the environment. It can be seen that, during the interaction with the environment, the contact force is much better regulated to the desired value of 10 N by the PI+NN control. This confirms that the manipulator is executing the test correctly.

Test 2. This test was conducted in the presence of a perturbation $F_{ext} = [f_{ext}, 0, f_{ext}, 0, 0, 0]$ with $f_{ext} = -1 - \sin(0.2t)$, in the sequel referred as 1 N perturbation, which is a force perpendicular to the tangent plane and opposite to the end-effector at the contact point. The results of the system response in position and force are provided in Figures 19–21, the effect of the perturbation can be observed after 0.5 s.

Figure 19 illustrates that the proposed controller is capable of maintaining the manipulator robot on the desired trajectory throughout the entire path, whereas with the classical PID the real trajectory largely deviates from the desired one once the disturbance

occurs. The figure also illustrates the controllers' performance in each of the x, y, z axes of the Cartesian plane.

Finally, the position and force errors are shown in Figures 20 and 21. The effect of the disturbance and the proposed controller's correction can be observed throughout the trajectory tracking.

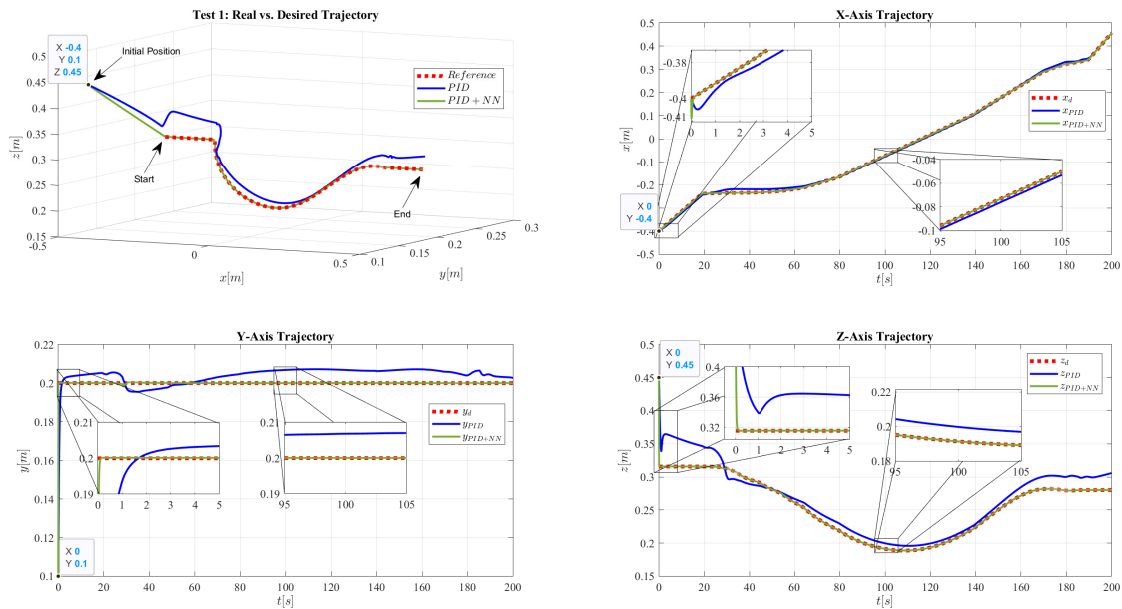


Figure 16. Trajectory tracking without perturbation. Reference (red dotted line) vs. PID (blue solid line) vs. PID+NN (green solid line).

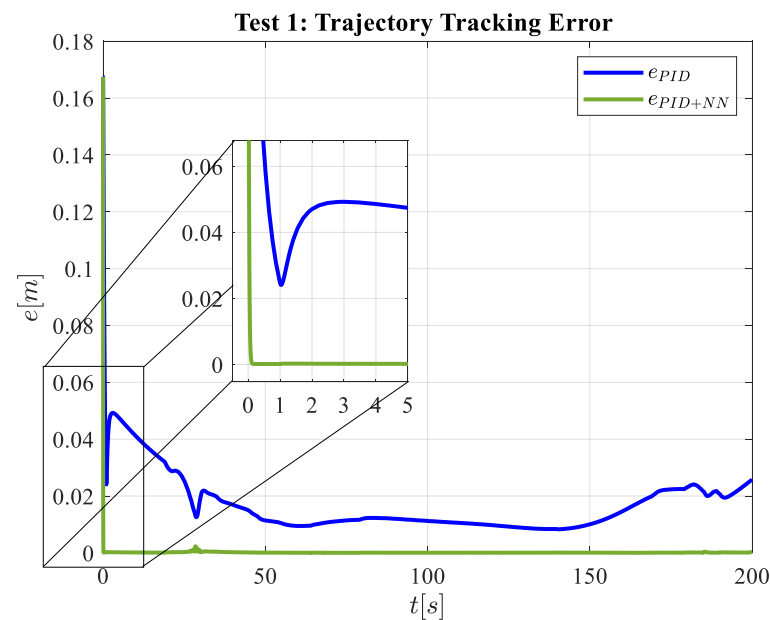


Figure 17. Trajectory tracking error without perturbation.

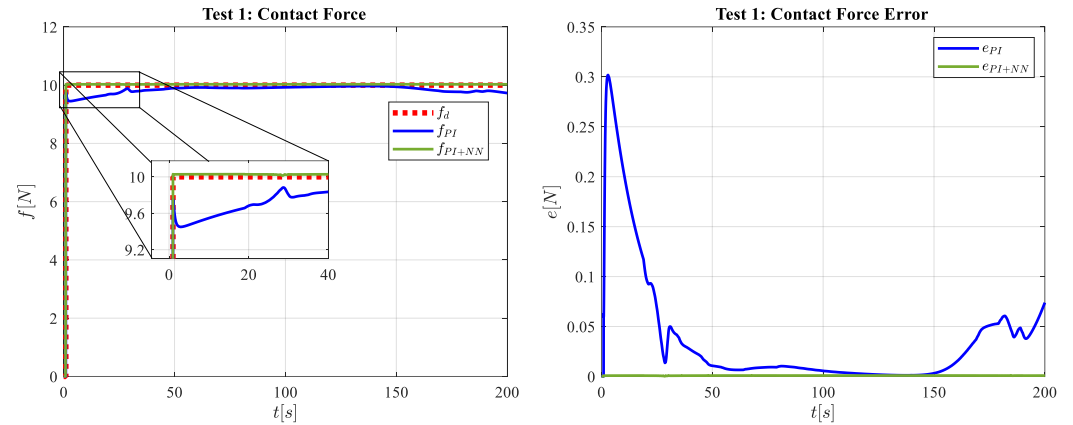


Figure 18. Force applied on the surface (left) and force error generated (right) without perturbation.

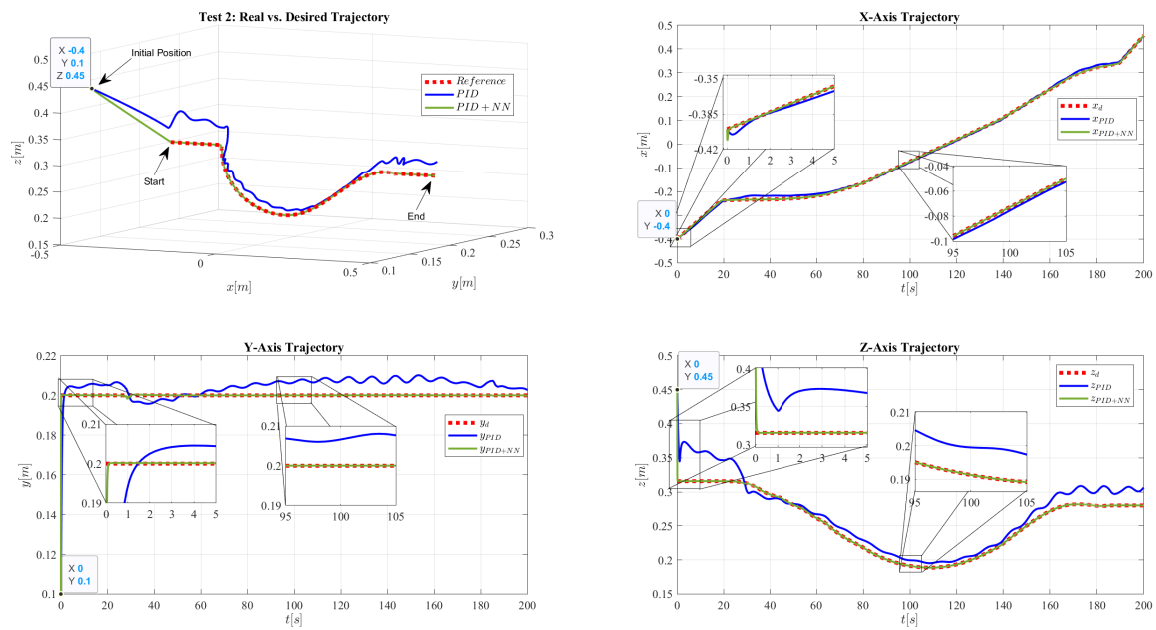


Figure 19. Trajectory tracking with variable perturbation of 1 N. Reference (red dotted line) vs. PID (blue solid line) vs. PID+NN (green solid line).

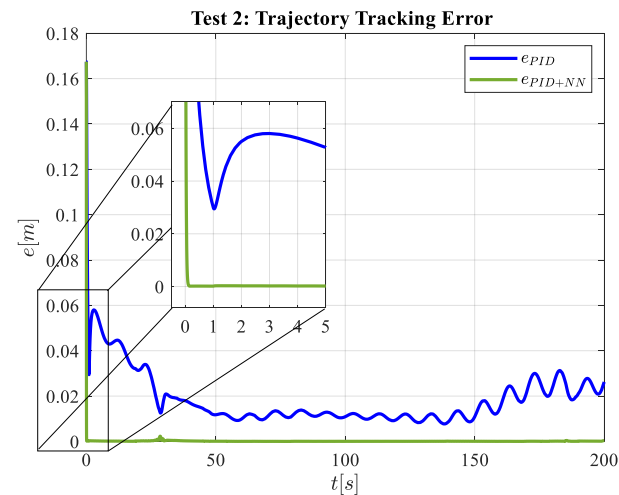


Figure 20. Trajectory tracking error with 1 N perturbation.

Test 3. For study purposes in this test, the perturbation magnitude is increased with respect to the Test 1, such that $f_{ext} = -5 - 5 \sin(0.2t)$, in the sequel referred as 5 N perturbation. The results of the system response are shown in Figures 22–24. It can be observed that the proposed controller is able to keep the manipulator robot in the desired trajectory and at a desired force value during the whole process with a minimum allowable error. In contrast, the classical PID is unable to approach the desired trajectory and force once the perturbation appears.

Test results demonstrate that the PID+NN controller is capable of tracking trajectories even in the presence of initial errors or variable disturbances, such as those that can cause unmodeled dynamics, tool changes, or high degrees of curvature on the surface. The robotic system is able to effectively handle the added error in a short response time, thanks to the correct compensation by the adaptive NN controller. The application of this intelligent control algorithm has yielded results suitable for future field testing.

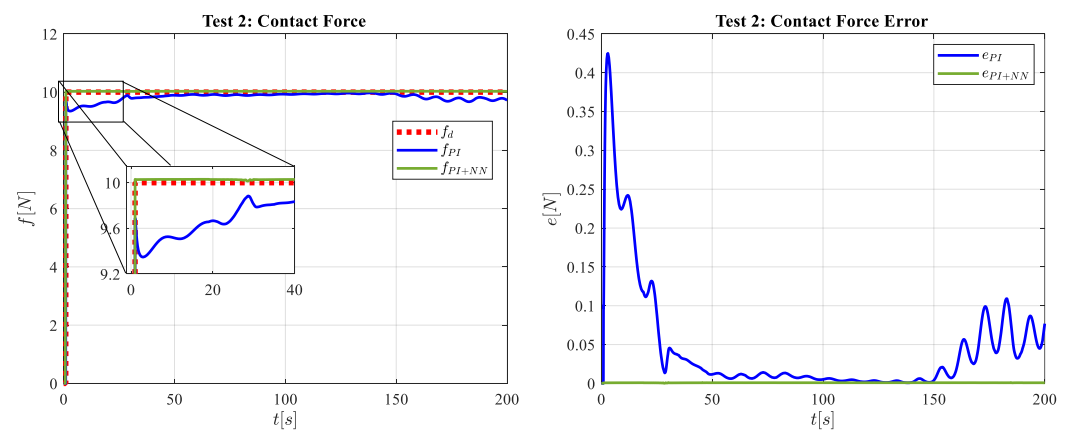


Figure 21. Force applied on the surface (left) and force error generated (right) with 1 N perturbation.

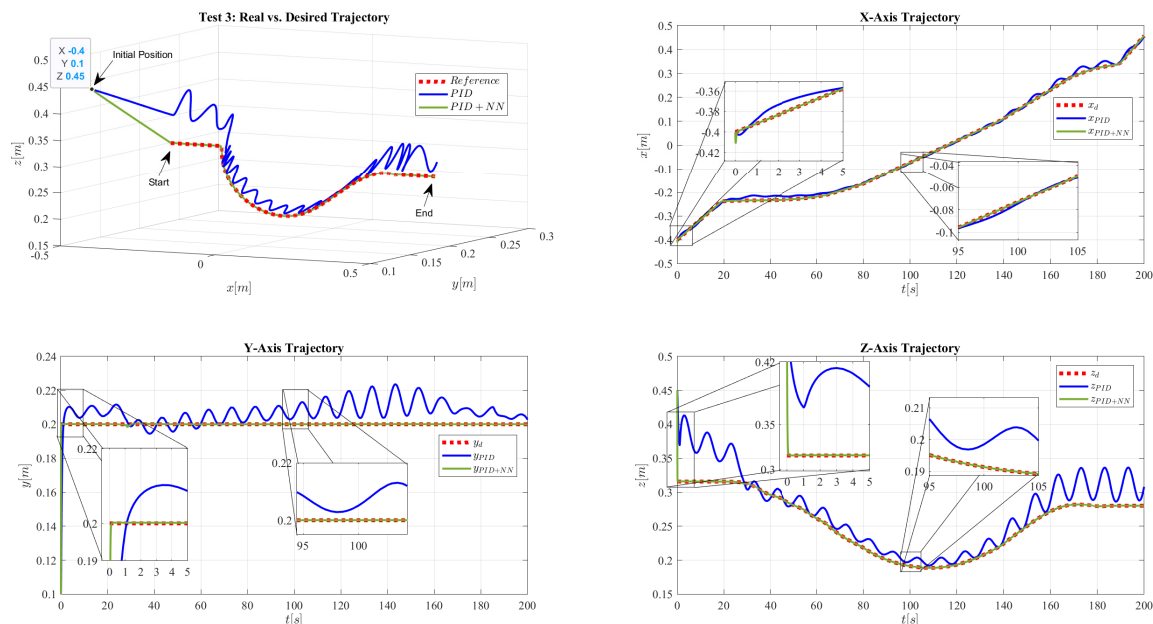


Figure 22. Trajectory tracking with variable perturbation of 5 N. Reference (red dotted line) vs. PID (blue solid line) vs. PID+NN (green solid line).

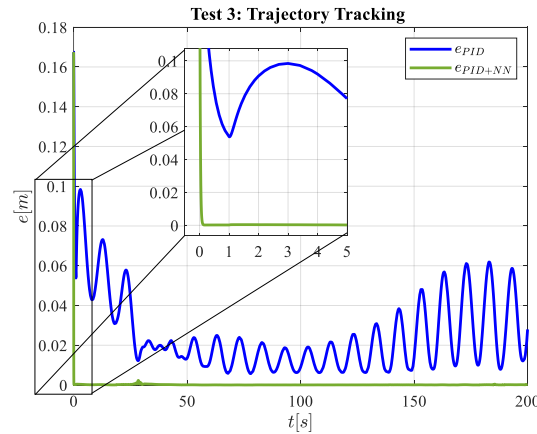


Figure 23. Trajectory tracking error with 5 N perturbation.

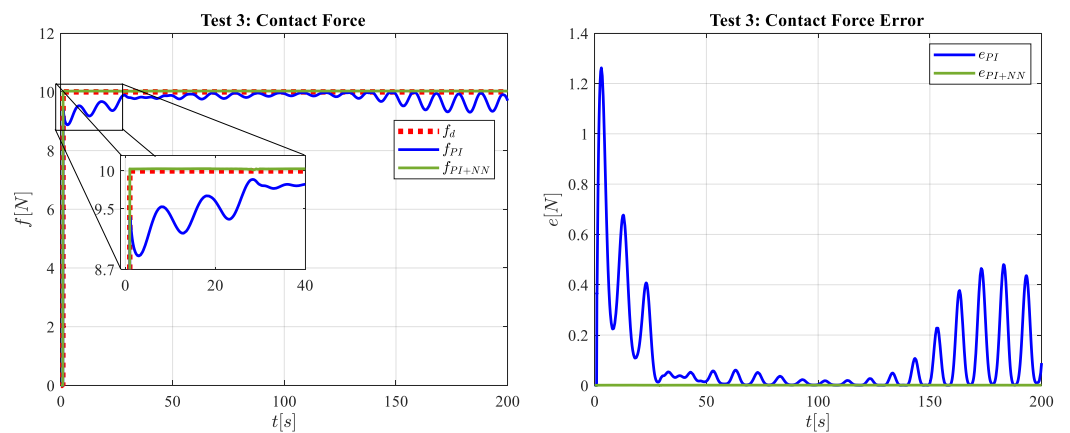


Figure 24. Force applied on the surface (left) and force error generated (right) with 5 N perturbation.

5.3. Performance Index

Finally, to evaluate the performance of the proposed controller, the root mean square error (RMSE) function was implemented in trajectory tracking.

$$RMSE_{xyz} = \sqrt{\frac{1}{n} \sum_{i=1}^n [(x_i - \hat{x})^2 + (y_i - \hat{y})^2 + (z_i - \hat{z})^2]} \quad (59)$$

where x_i , y_i , z_i are the actual values of the response variable in the observations made; $\hat{x}$, $\hat{y}$, $\hat{z}$ correspond to the predicted values of the same response variable in those same observations and n is the number of observations [67].

Figure 25 presents the results of the three tests of scenario two. It can be seen that the proposed control algorithm outperforms the other algorithms in each of the tests performed. On the left side of Figure 25, the response bars of the conventional PID control are observed. It can be seen that without disturbance (green bar), the control follows the trajectory with a deviation of 2.11 cm with respect to the reference; as the disturbance increases (orange and red bars), the error increases up to 63% with respect to the first test performed.

The bars on the right show the performance of the proposed control in this paper; excellent performance is observed under ideal conditions, the control follows the trajectory with a deviation of 0.02573 cm with respect to the reference, once the 1 N disturbance appears, the control system responds much better with respect to classical PID, reducing the error 5% with respect to the first test. In test 3, where the disturbance increases to 5 N, the performance of the proposed control is still superior to that of the conventional control system, achieving a reduction in RMSE of up to 96% with respect to the conventional PID. These results demonstrate the importance of adaptive neural control in interaction tests for a robotic system.

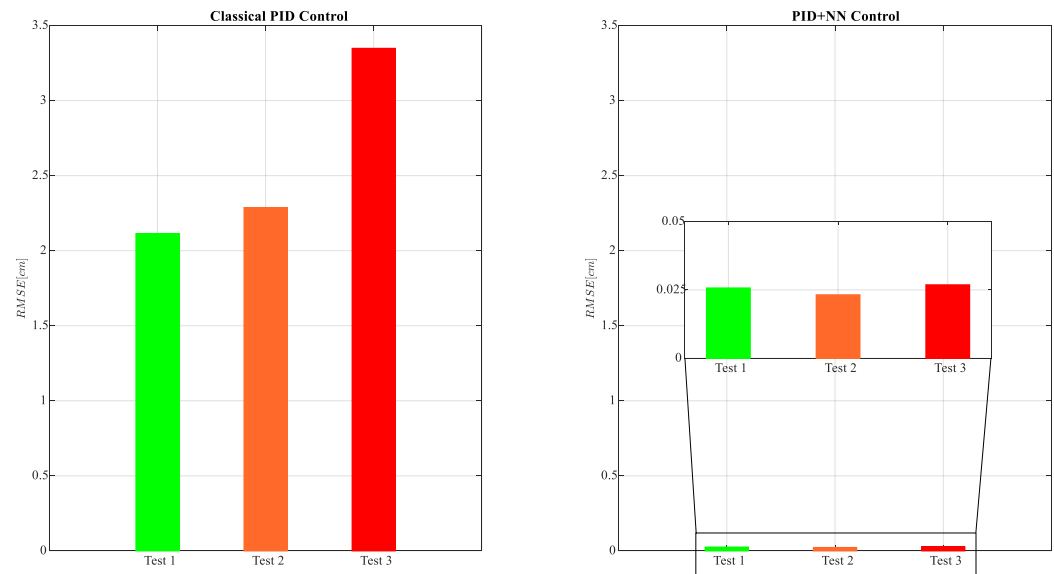


Figure 25. RMSE: classical PID control (left) vs. PID+NN control (right).

6. Conclusions

This work presents a novel control architecture for force/position control in n DoF manipulator robots. The architecture combines a neural network with a classical PID/PI controllers, offering simplicity and ease of implementation. A Lyapunov function guarantees the exponential stability of the non-linear system, ensuring that the closed-loop system state converges exponentially to the origin. The proposed approach was subjected to rigorous testing in two simulation scenarios, using a 6 DoF robot and a complex surface. This demonstrated the high performance and robustness of the proposed approach and confirmed the theoretical statements in very demanding dynamic motions. Another highlight of this study was the ability of the neural controller to adapt to the environment without relying on knowledge of the model parameters and structure for its real-time design and implementation. The neural network weight coefficients were updated online using position and force data while the controller was applied to the system, without prior training of the neural network weights. For comparative purposes, a study was conducted between the conventional PID and the proposed adaptive PID+NN. Two criteria were used to evaluate the performance of each controller: the RMSE in trajectory tracking and settling time. The results demonstrated that the proposed controller achieved the best performance and throughput. The RMSE values demonstrate that the adaptive PID+NN action achieved the best performance. Therefore, the combination of both controllers results in a more robust and versatile controller. The implementation and tracking of trajectories on a real platform will be addressed in future work.

Author Contributions: Conceptualization, J.F.V.-T., J.A.E.-T. and R.H.-A.; methodology, J.F.V.-T., J.A.E.-T., E.A.F.-U. and D.B.-P.; validation, J.F.V.-T., J.A.E.-T., R.H.-A. and E.A.F.-U.; formal analysis, J.F.V.-T., J.A.E.-T. and E.A.F.-U.; investigation, J.F.V.-T., D.B.-P., E.A.F.-U. and J.R.M.-M.; resources, E.A.F.-U., J.A.E.-T., R.H.-A. and D.B.-P.; writing—original draft preparation, J.F.V.-T., J.A.E.-T. and E.A.F.-U.; writing—review and editing, J.A.E.-T., E.A.F.-U. and R.H.-A.; supervision, R.H.-A., J.A.E.-T. and E.A.F.-U.; project administration D.B.-P., E.A.F.-U. and J.R.M.-M.; funding acquisition, E.A.F.-U. and J.A.E.-T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Center for Engineering and Industrial Development (CIDESI) Querétaro through the project QID029.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to acknowledge the technical support provided by Center for Engineering and Industrial Development (CIDESI) Querétaro. The authors would like to

thanks the technical support provided by Polytechnic University of Queretaro (UPQ). J.F.V.-T. thanks CONAHCYT for their support during graduate studies.

Conflicts of Interest: The authors declare no competing or conflict of interest occurred in this work.

Appendix A

Algorithm A1 Pseudo-code for complex trajectory generation

Input: New file of type “*.stl” extension.

Output: Desired trajectory matrix T_d with coordinates x, y, z .

```

1: function S=DESIRED_TRAJECTORIES(NameFile)
2:   points = readfile(NameFile);           ▷ NameFile = “./SectionAlabeTurbine.stl”
3:   X=points(:,1);                         ▷ Number of points on X-axis
4:   Y=points(:,2);                         ▷ Number of points on Y-axis
5:   Z=points(:,3);                         ▷ Number of points on Z-axis
6:    $T_{xyz} = [X \ Y \ Z]$ ;                 ▷ Trajectory matrix to interpolate
7:   Nmax=a;                               ▷ Maximum number of points per trajectory
8:   function S1=SORT_COORDINATES_XYZ( $T_{xyz}$ )
9:     [row, col] = size( $T_{xyz}$ );
10:    for  $i = 1$  to length(row) do
11:      while ( $i > 0$ ) & ( $T_{xyz}(i, 2) < T_{xyz}(i - 1, 2)$ ) do
12:        Sort the matrix  $T_{xyz}$  as a function of the terms of Y;
13:      end while
14:    end for
15:  end function
16:  function S2=GROUP_TRAJECTORIES( $T_{xyz}$ )
17:     $vu = \text{unique}(T_{xyz}(:, 2))$ ;
18:    for  $i = 1$  to length( $vu$ ) do
19:      Form matrix trajectory:  $T = [T_1 \ T_2 \ T_3 \ \dots \ T_n]$ ;
20:    end for
21:  end function
22:  function S3=REMOVE_DUPLICATE_COORDINATES( $T$ )
23:    Remove duplicate coordinates in  $T$ ;
24:  end function
25:  function S4=SORT_COORDINATES_XYZ( $T$ )
26:    [row, col] = size( $T$ );
27:    for  $i = 1$  to length(row) do
28:      while ( $i > 0$ ) & ( $T(i, 1) < T(i - 1, 1)$ ) do
29:        Sort the matrix  $T$  as a function of the terms of X;
30:        Compute new matrix to interpolate:  $T_{new}$ ;
31:      end while
32:    end for
33:  end function
34:  function S5=CUBIC_INTERPOLATOR( $T_{new}, Nmax$ )
35:    Apply the cubic interpolator to matrix  $T_{new}$ ;
36:    Compute the desired trajectory matrix:  $T_d = [T_1 \ T_2 \ T_3 \ \dots \ T_n]$ ;
37:    if  $T_d$  is optimal then
38:       $S = T_d$ ;
39:    else
40:      return S5
41:    end if
42:  end function
43: end function

```

References

1. Marsh, G. Automating aerospace composites production with fibre placement. *Reinf. Plast.* **2011**, *55*, 32–37. [\[CrossRef\]](#)
2. Zhang, L.; Wang, X.; Pei, J.; Zhou, Y. Review of automated fibre placement and its prospects for advanced composites. *J. Mater. Sci.* **2020**, *55*, 7121–7155. [\[CrossRef\]](#)
3. Lukaszewicz, D.H.J.; Ward, C.; Potter, K.D. The engineering aspects of automated prepreg layup: History, present and future. *Compos. Part Eng.* **2012**, *43*, 997–1009. [\[CrossRef\]](#)
4. Brasington, A.; Sacco, C.; Halbritter, J.; Wehbe, R.; Harik, R. Automated fiber placement: A review of history, current technologies, and future paths forward. *Compos. Part Open Access* **2021**, *6*, 100182. [\[CrossRef\]](#)
5. Crosky, A.; Grant, C.; Kelly, D.; Legrand, X.; Pearce, G. 4—Fibre placement processes for composites manufacture. In *Advances in Composites Manufacturing and Process Design*; Boisse, P., Ed.; Woodhead Publishing: Sawston, UK, 2015; pp. 79–92. [\[CrossRef\]](#)
6. Yan, L.; Chen, Z.C.; Shi, Y.; Mo, R. An accurate approach to roller path generation for robotic fibre placement of free-form surface composites. *Robot. Comput.-Integr. Manuf.* **2014**, *30*, 277–286. [\[CrossRef\]](#)
7. Zhang, W.; Liu, F.; Jiang, T.; Yi, M.; Chen, W.; Ding, X. Overview of current design and analysis of potential theories for automated fibre placement mechanisms. *Chin. J. Aeronaut.* **2022**, *35*, 1–13. [\[CrossRef\]](#)
8. Hu, P.; Zhou, H.; Chen, J.; Lee, C.; Tang, K.; Yang, J.; Shen, S. Automatic generation of efficient and interference-free five-axis scanning path for free-form surface inspection. *Comput.-Aided Des.* **2018**, *98*, 24–38. [\[CrossRef\]](#)
9. Belnoue, J.P.H.; Mesogitis, T.; Nixon-Pearson, O.J.; Kratz, J.; Ivanov, D.S.; Partridge, I.K.; Potter, K.D.; Hallett, S.R. Understanding and predicting defect formation in automated fibre placement pre-preg laminates. *Compos. Part Appl. Sci. Manuf.* **2017**, *102*, 196–206. [\[CrossRef\]](#)
10. Lukaszewicz, D.H.J.; Potter, K.D.; Eales, J. A concept for the in situ consolidation of thermoset matrix prepreg during automated lay-up. *Compos. Part Eng.* **2013**, *45*, 538–543. [\[CrossRef\]](#)
11. Jiang, J.; He, Y.; Wang, H.; Ke, Y. Modeling and experimental validation of compaction pressure distribution for automated fiber placement. *Compos. Struct.* **2021**, *256*, 113101. [\[CrossRef\]](#)
12. Gao, J.; Pashkevich, A.; Caro, S. Optimization of the robot and positioner motion in a redundant fiber placement workcell. *Mech. Mach. Theory* **2017**, *114*, 170–189. [\[CrossRef\]](#)
13. Zhang, L.; Wang, X.; Pei, J.; Nian, C. Adaptive Path Planning of Fiber Placement Based on Improved Method of Mesh Dynamic Representation. *Appl. Compos. Mater.* **2019**, *26*, 785–803. [\[CrossRef\]](#)
14. Uhart, M.; Patrouix, O.; Aoustin, Y. Improving manufacturing of aeronautical parts with an enhanced industrial Robotised Fibre Placement Cell using an external force-vision scheme. *Int. J. Interact. Des. Manuf.* **2016**, *10*, 15–35. [\[CrossRef\]](#)
15. Zhang, X.; Zhou, H.; Liu, J.; Ju, Z.; Leng, Y.; Yang, C. A practical PID variable stiffness control and its enhancement for compliant force-tracking interactions with unknown environments. *Sci. China Technol. Sci.* **2023**, *66*, 2882–2896. [\[CrossRef\]](#)
16. Wahballa, H.; Duan, J.; Dai, Z. Constant force tracking using online stiffness and reverse damping force of variable impedance controller for robotic polishing. *Int. J. Adv. Manuf. Technol.* **2022**, *121*, 5855–5872. [\[CrossRef\]](#)
17. Amersdorfer, M.; Kappey, J.; Meurer, T. Real-time freeform surface and path tracking for force controlled robotic tooling applications. *Robot. Comput.-Integr. Manuf.* **2020**, *65*, 101955. [\[CrossRef\]](#)
18. Song, X.; Li, B.; Xu, W.; Li, Z. Coupled Force–Position Control for Dynamic Contact Force Tracking in Uncertain Environment. *Actuators* **2022**, *11*, 150. [\[CrossRef\]](#)
19. Gan, Y.; Duan, J.; Chen, M.; Dai, X. Multi-Robot Trajectory Planning and Position/Force Coordination Control in Complex Welding Tasks. *Appl. Sci.* **2019**, *9*, 924. [\[CrossRef\]](#)
20. Wahballa, H.; Duan, J.; Wang, W.; Dai, Z. Experimental Study of Robotic Polishing Process for Complex Violin Surface. *Machines* **2023**, *11*, 147. [\[CrossRef\]](#)
21. Xu, X.; Zhu, D.; Zhang, H.; Yan, S.; Ding, H. Application of novel force control strategies to enhance robotic abrasive belt grinding quality of aero-engine blades. *Chin. J. Aeronaut.* **2019**, *32*, 2368–2382. [\[CrossRef\]](#)
22. Ohhira, T.; Yokota, K.; Tatsumi, S.; Murakami, T. A Robust Hybrid Position/Force Control Considering Motor Torque Saturation. *IEEE Access* **2021**, *9*, 34515–34528. [\[CrossRef\]](#)
23. Jinjun, D.; Yahui, G.; Ming, C.; Xianzhong, D. Symmetrical adaptive variable admittance control for position/force tracking of dual-arm cooperative manipulators with unknown trajectory deviations. *Robot. Comput.-Integr. Manuf.* **2019**, *57*, 357–369. [\[CrossRef\]](#)
24. Dai, J.; Zhang, Y.; Deng, H. Novel Voltage-Based Weighted Hybrid Force/Position Control for Redundant Robot Manipulators. *Electronics* **2022**, *11*, 179. [\[CrossRef\]](#)
25. Wang, H.; Xie, Y. Adaptive Jacobian position/force tracking control of free-flying manipulators. *Robot. Auton. Syst.* **2009**, *57*, 173–181. [\[CrossRef\]](#)
26. Jiang, J.; Luo, J.; Wang, H.; Tang, X.; Nian, F.; Qi, L. Force Tracking Control Method for Robotic Ultrasound Scanning System under Soft Uncertain Environment. *Actuators* **2024**, *13*, 62. [\[CrossRef\]](#)
27. Rayguru, M.M.; Elara, M.R.; Gómez, B.F.; Ramalingam, B. A Time Delay Estimation Based Adaptive Sliding Mode Strategy for Hybrid Impedance Control. *IEEE Access* **2020**, *8*, 155352–155361. [\[CrossRef\]](#)
28. Gracia, L.; Solanes, J.E.; Muñoz-Benavent, P.; Valls Miro, J.; Perez-Vidal, C.; Tornero, J. Adaptive Sliding Mode Control for Robotic Surface Treatment Using Force Feedback. *Mechatronics* **2018**, *52*, 102–118. [\[CrossRef\]](#)

29. Vázquez-Alberto, L.F.; Arteaga, M.A. A continuous terminal sliding mode algorithm for robot manipulators: an application to force control. *Int. J. Control.* **2023**, *96*, 2812–2826. [\[CrossRef\]](#)
30. Al-Dujaili, A.Q.; Falah, A.; Humaidi, A.J.; Pereira, D.A.; Ibraheem, I.K. Optimal super-twisting sliding mode control design of robot manipulator: Design and comparison study. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1729881420981524. [\[CrossRef\]](#)
31. Ahmed, S.; Wang, H.; Tian, Y. Adaptive Fractional High-order Terminal Sliding Mode Control for Nonlinear Robotic Manipulator under Alternating Loads. *Asian J. Control.* **2021**, *23*, 1900–1910. [\[CrossRef\]](#)
32. Gutiérrez-Giles, A.; Arteaga-Pérez, M.A.; Sira-Ramírez, H. Control de Fuerza de Robots Manipuladores Basado en Observadores Proporcionales Integrales Generalizados. *Rev. Iberoam. Automática Informática Ind. Riad* **2016**, *13*, 238–246. [\[CrossRef\]](#)
33. Piltan, F.; Kim, C.H.; Kim, J.M. Advanced Adaptive Fault Diagnosis and Tolerant Control for Robot Manipulators. *Energies* **2019**, *12*, 1281. [\[CrossRef\]](#)
34. Cao, P.; Gan, Y.; Dai, X. Finite-Time Disturbance Observer for Robotic Manipulators. *Sensors* **2019**, *19*, 1943. [\[CrossRef\]](#)
35. Sánchez-Sánchez, P.; Arteaga-Pérez, M.A. Improving force tracking control performance in cooperative robots. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1729881417708969. [\[CrossRef\]](#)
36. Tuan, D.M.; Hieu, P.D. Adaptive Position/Force Control for Robot Manipulators Using Force and Velocity Observer. *J. Electr. Eng. Technol.* **2019**, *14*, 2575–2582. [\[CrossRef\]](#)
37. Arteaga, M.A.; Gutierrez-Giles, A. On the robustness of force estimation methods for robot manipulators: An experimental study. *J. Frankl. Inst.* **2023**, *360*, 11705–11735. [\[CrossRef\]](#)
38. Karami, A.; Sadeghian, H.; Keshmiri, M.; Oriolo, G. Force, orientation and position control in redundant manipulators in prioritized scheme with null space compliance. *Control. Eng. Pract.* **2019**, *85*, 23–33. [\[CrossRef\]](#)
39. Peng, J.; Yang, Z.; Ma, T. Position/Force Tracking Impedance Control for Robotic Systems with Uncertainties Based on Adaptive Jacobian and Neural Network. *Complexity* **2019**, *2019*, 1406534. [\[CrossRef\]](#)
40. Ding, S.; Peng, J.; Zhang, H.; Wang, Y. Neural network-based adaptive hybrid impedance control for electrically driven flexible-joint robotic manipulators with input saturation. *Neurocomputing* **2021**, *458*, 99–111. [\[CrossRef\]](#)
41. Luan, F.; Na, J.; Huang, Y.; Gao, G. Adaptive neural network control for robotic manipulators with guaranteed finite-time convergence. *Neurocomputing* **2019**, *337*, 153–164. [\[CrossRef\]](#)
42. An, H.; Ye, C.; Yin, Z.; Lin, W. Neural Adaptive Impedance Control for Force Tracking in Uncertain Environment. *Electronics* **2023**, *12*, 640. [\[CrossRef\]](#)
43. Lin, S.T.; Tsai, H.C. Impedance Control with On-Line Neural Network Compensator for Dual-Arm Robots. *J. Intell. Robot. Syst.* **1997**, *18*, 87–104. [\[CrossRef\]](#)
44. Gómez-Espinosa, A.; Castro Sundin, R.; Loidi Eguren, I.; Cuan-Urquiza, E.; Treviño-Quintanilla, C.D. Neural Network Direct Control with Online Learning for Shape Memory Alloy Manipulators. *Sensors* **2019**, *19*, 2576. [\[CrossRef\]](#)
45. Liu, M.; He, L.; Hu, B.; Li, S. Recurrent neural network with noise rejection for cyclic motion generation of robotic manipulators. *Neural Netw.* **2021**, *138*, 164–178. [\[CrossRef\]](#)
46. Su, H.; Hu, Y.; Karimi, H.R.; Knoll, A.; Ferrigno, G.; De Momi, E. Improved recurrent neural network-based manipulator control with remote center of motion constraints: Experimental results. *Neural Netw.* **2020**, *131*, 291–299. [\[CrossRef\]](#)
47. Xu, Z.; Li, S.; Zhou, X.; Cheng, T. Dynamic neural networks based adaptive admittance control for redundant manipulators with model uncertainties. *Neurocomputing* **2019**, *357*, 271–281. [\[CrossRef\]](#)
48. Wang, Y. Robot algorithm based on neural network and intelligent predictive control. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 6155–6166. [\[CrossRef\]](#)
49. Rossomando, F.G.; Serrano, E.; Soria, C.M.; Scaglia, G. Neural Dynamics Variations Observer Designed for Robot Manipulator Control Using a Novel Saturated Control Technique. *Math. Probl. Eng.* **2020**, *2020*, 3240210. [\[CrossRef\]](#)
50. Ferguene, F.; Toumi, R. Dynamic External Force Feedback Loop Control of a Robot Manipulator Using a Neural Compensator—Application to the Trajectory Following in an Unknown Environment. *Int. J. Appl. Math. Comput. Sci.* **2009**, *19*, 113–126. [\[CrossRef\]](#)
51. Parra-Vega, V.; García-Rodríguez, R.; Ruiz-Sanchez, F. Second Order Sliding Mode Adaptive Neurocontrol for Robot Arms with Finite Time Convergence. *Ifac Proc. Vol.* **2005**, *38*, 235–240. [\[CrossRef\]](#)
52. Sun, C.; Li, G.; Xu, J. Adaptive neural network terminal sliding mode control for uncertain spatial robot. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419894065. [\[CrossRef\]](#)
53. Truong, H.V.A.; Tran, D.T.; Ahn, K.K. A Neural Network Based Sliding Mode Control for Tracking Performance with Parameters Variation of a 3-DOF Manipulator. *Appl. Sci.* **2019**, *9*, 23. [\[CrossRef\]](#)
54. Zhang, W.; Ye, X.; Jiang, L.; Zhu, Y.; Ji, X.; Hu, X. Output feedback control for free-floating space robotic manipulators base on adaptive fuzzy neural network. *Aerosp. Sci. Technol.* **2013**, *29*, 135–143. [\[CrossRef\]](#)
55. Parra-Vega, V.; García-Rodríguez, R.; Armendariz, J. Neurofuzzy self-tuning of the dissipation rate gain for model-free force-position exponential tracking of robots. *Neurocomputing* **2016**, *171*, 209–219. [\[CrossRef\]](#)
56. Kelly, R.; Davila, V.S.; Loria, A. *Control of Robot Manipulators in Joint Space*; Springer: London, UK, 2005. [\[CrossRef\]](#)
57. Pliego-Jiménez, J.; Arteaga-Pérez, M.A. Adaptive position/force control for robot manipulators in contact with a rigid surface with uncertain parameters. *Eur. J. Control.* **2015**, *22*, 1–12. [\[CrossRef\]](#)
58. Li, J.; Guan, Y.; Chen, H.; Wang, B.; Zhang, T.; Hong, J.; Wang, D. Real-time normal contact force control for robotic surface processing of workpieces without a priori geometric model. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 2537–2551. [\[CrossRef\]](#)

59. SCHUNK. Force/Torque Sensor FT Mini58. 2024. Available online: https://schunk.com/de/en/automation-technology/force/torque-sensors/ft/c/PGR_1680 (accessed on 2 May 2024).
60. Cuevas, E.; Luque, A.; Escobar, H. *Computational Methods with MATLAB*; Springer: Berlin/Heidelberg, Germany, 2024. [CrossRef]
61. Madani, M.; Moallem, M. Hybrid position/force control of a flexible parallel manipulator. *J. Frankl. Inst.* **2011**, *348*, 999–1012. [CrossRef]
62. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. *Force Control*; Springer: London, UK, 2009; pp. 363–405. [CrossRef]
63. Lewis, F.L.; Yesildirak, A.; Jagannathan, S. *Neural Network Control of Robot Manipulators and Nonlinear Systems*; Taylor & Francis, Inc.: New York, NY, USA, 1998.
64. Rashid, T. *Make Your Own Neural Network*; CreateSpace Independent Publishing Platform: North Charleston, SC, USA, 2016; Volume 29.
65. Ponce, A.N.; Behar, A.A.; Hernández, A.O.; Sitar, V.R. Neural Networks for Self-tuning Control Systems. *Acta Polytech.* **2004**, *44*, 49–52. [CrossRef]
66. Memar, A.H.; Esfahani, E.T. Modeling and Dynamic Parameter Identification of the SCHUNK Powerball Robotic Arm. In *Volume 5C: 39th Mechanisms and Robotics Conference; V05CT08A024*; American Society of Mechanical Engineers: New York, NY, USA, 2015. [CrossRef]
67. Agüera-Vega, F.; Carvajal-Ramírez, F.; Martínez-Carricondo, P. Accuracy of Digital Surface Models and Orthophotos Derived from Unmanned Aerial Vehicle Photogrammetry. *J. Surv. Eng.* **2017**, *143*. [CrossRef]

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