IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 December 2025, accepted 18 January 2026, date of publication 21 January 2026, date of current version 3 February 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3656712

RESEARCH ARTICLE

Adaptive Neural Network Control for Trajectory Tracking and Vibration Suppression in Flexible-Link Parallel Manipulators

RENAN SANCHES GERONEL, FÁBIO LÚCIO FELIX, AND MAÍRA MARTINS DA SILVA
Department of Mechanical Engineering, São Carlos School of Engineering, University of São Paulo, São Carlos, São Paulo, Brazil

Corresponding author: Maíra Martins da Silva (mairams@sc.usp.br)

This work was supported in part by São Paulo Research Foundation (FAPESP) under Grant 18/21336-0 and Grant 23/07834-6, in part by the Coordination for the Improvement of Higher Education Personnel under Grant 001, and in part by the National Council for Scientific and Technological Development (CNPq) under Grant 303736/2024-0.

ABSTRACT Robotic manipulators are widely employed in applications that demand both high precision and high speed. Among the different configurations, parallel manipulators with flexible links have emerged as a promising alternative due to their higher energy efficiency, improved compliance, lighter structures, and smaller actuators. However, the reduction in mass increases the number of degrees of freedom, which makes mathematical modeling more complex and also induces undesired vibrations in the mechanical system. In this sense, an adaptive control based on a neural network is designed to both improve the tracking trajectory and attenuate the negative effects mainly caused by the low thickness of the links. The dataset to train the ANN is generated based on different trajectory configurations of the prototype. The end-effector is initially placed at different locations of the workspace to investigate the flexible characteristics over distinct scenarios. Once tested, additional signals (not used previously in the training phase) are randomly selected to validate the proposed strategy in experimental trials. The control strategy is then calculated by summing the pose loop, based on a PID strategy, and the adaptive control law, based on the ANN model. Experimental results demonstrate that the proposed control approach achieves at least a 27.70% reduction compared to the Pose loop scheme, highlighting its effectiveness in enhancing tracking accuracy in flexible-link parallel manipulators. This proposed control strategy is used to compensate in real-time for the overall dynamics regardless of the trajectory, model uncertainties, or undesired effects.

INDEX TERMS Parallel manipulators with flexible links, adaptive control based on neural network, real-time compensation, vibration attenuation.

I. INTRODUCTION

Robotic manipulators have attracted significant attention due to their ability to perform high-precision and high-speed tasks in diverse applications, such as semiconductor manufacturing, the aerospace industry, and automatic micro-assembly [1], [2]. Two main types of robotic manipulators are currently employed in these applications: serial and parallel manipulators. While serial manipulators often present large mass and limitations in mobility and dynamic response, mainly due to actuator torque constraints, parallel manipulators (PMs) offer several advantages, including higher energy efficiency, better compliance, lighter structures, and the use of smaller actuators. The adoption of PMs with flexible links is accompanied by continuous mass reduction since they represent a promising design alternative capable of achieving higher speed, acceleration, and payload-to-weight ratio while consuming less energy compared to PMs with rigid links [3]. Nevertheless, this reduction in mass may induce undesired vibrations in the mechanical system due to inertial and coupling effects, which negatively impact end-effector accuracy, settling time, and overall system stability [4]. Consequently, these challenges have motivated the continuous development of robust control strategies to

The associate editor coordinating the review of this manuscript and approving it for publication was Mouquan Shen.

address with complex coupled dynamic systems and vibration suppression for parallel manipulators with flexible links [5], [6], [7].

Research on flexible links has predominantly focused on multi-link serial manipulators and cable-driven robots [8], [9], [10], [11], while investigations into the modeling of PMs with flexible links remain scarce. This scarcity is mainly due to the complexity of the equations governing flexible links and the challenges of designing robust and efficient control techniques [12]. Moreover, PMs are sensitive to structural errors, and the strong coupling between kinematic chains and multi–degree-of-freedom motion makes overall motion accuracy and error compensation a challenging task [13], [14], [15]. Alternatively, model-free control strategies have recently been investigated for flexible robotic manipulators. Bengoa et al. [16] explore a virtual sensing approach, whereas Zake et al. [17] assess a pose-based visual servoing control strategy. Moreover, Colombo and da Silva [18] investigated two hybrid model-based control approaches for a flexible parallel manipulator. Both strategies are designed based on two loops: a PID feedback loop that employs encoder measurement signals and an inverse kinematic model (IKM), and an LQR/LQG feedback loop that estimates the pose of the end-effector. Note that the IKM is derived assuming that the parallel manipulator is composed of rigid links. This second loop estimates the end-effector pose using images acquired by a camera, while the other alternative investigates a reduced-order finite element model. Experimental results reveal that the proposed strategies can reduce steady-state errors and attenuate vibrating responses, endorsing the contribution of hybrid control strategies.

Zhang et al. [19] assess the design of a parallel compliant leg (PCL) mechanism that integrates passive elastic components into a walker to cooperate with the actuated legs of an exoskeleton. The energy efficiency of this mechanical system is achieved by reducing the active torque demands by storing and releasing mechanical energy during the gait cycle. Numerical and experimental results show that the PCL can reduce the peak torque of 70%, while decreasing the total energy consumption by 70.9%. Similarly, Cheng et al. [20] investigate a dual-impulse vibration suppression controller for a large-size flexible hydraulic manipulator. The proposed strategy is designed to enable an online trade-off between vibration suppression and precise positioning in a specific pumping posture. One advantage of the proposed strategy is the lack of need for mathematical modeling to obtain the vibration suppression controller. Consequently, an improvement in stability is achieved, as only pressure feedback is required to mitigate the vibration and ensure an accurate position of the end-effector. Therefore, several robust strategies have been investigated for flexible parallel manipulators to enhance energy efficiency, allowing the use of smaller actuators and lightweight flexible links, while also coping with model uncertainties and suppressing undesired oscillations.

However, since traditional model-free and model-based techniques struggle to deliver the required precision and real-time capability, alternative approaches are gaining continuous importance in the industrial and research fields. Machine Learning (ML), including Artificial Neural Networks (ANN), has emerged as a promising alternative to these limitations, offering powerful capabilities for modeling and optimizing complex robotic systems [21], [22], [23]. Yu [24] investigates the design of an artificial neural network to learn the kinematic model error of a parallel robot. The proposed strategy demonstrates its effectiveness in both numerical and experimental analyses. Moreover, Yu [25] assesses a hybrid methodology, based on a back-propagation (BP) neural network and radial basis function (RBF) neural network, and a control module to compensate for the pose errors of a parallel manipulator caused mainly by non-geometrical parameter errors. Experimental results reveal that the proposed strategy improves the pose accuracy of the parallel manipulator. Li and Wang [26] have improved the conventional Proportional Integral Derivative (PID) control by including Diagonal Recurrent Neural Networks (DRNN) to enhance the adaptivity of the mechanical system. The neural network PID shows a positive reduction in tracking errors and, consequently, a trajectory tracking performance improvement of the parallel robotic system.

Given this context, adaptive neural network controllers can be used to cope with environmental changes and model uncertainties in real time [32]. In practice, several assumptions are often made when designing efficient control strategies and mathematical models for parallel manipulators, which in turn require compensations for these uncertainties. Rahmani and Belkheiri [28] assess the design of an adaptive neural network output-feedback control strategy for flexible multi-link robotic manipulators. NN is used to approximate and eliminate the effects of uncertainties due to robotic flexibility and vibration on tracking performance. Nguyen et al. [29] investigate the design of an adaptive finite-time neural network parallel manipulator. Simulations reveal that the proposed control outperforms conventional strategies in terms of tracking accuracy and convergence speed. Zhang et al. [30] investigate the design of an adaptive sliding mode control algorithm based on a neural network for improving the tracking performance of the parallel manipulator and also attenuating the undesired vibration caused by flexible links. Compared to a traditional PID, the proposed strategy shows superior responses in terms of system stability and trajectory accuracy. Moreover, Yu et al. [31] assess the use of a radial basis function neural network (RBFNN) for controlling a flexible planar parallel manipulator. The RBFNN is designed based on acceleration feedback to both suppress the undesired vibrations and improve the trajectory tracking of the mechanical system.

In addition, Liu et al. [32] investigate a fault-tolerant control scheme for networked flexible manipulators subject to actuator failures, input saturations, and external

disturbances. The effect of input saturation is mitigated through a second-order auxiliary system, while nonlinear uncertainties are approximated using adaptive neural networks. In addition, a boundary local-angle interaction protocol is designed under a directed communication graph to achieve distributed coordination among the manipulators. Li and Liu [33] assess the use of a neural-network-based adaptive fault-tolerant vibration control strategy for a flexible manipulator subject to model uncertainties, external disturbances, and potential actuator faults. A neural network is used to approximate unknown nonlinear dynamics, while adaptive laws adjust the controller parameters to compensate for faults and ensure stable performance. The results reveal that the proposed method achieves positive robustness, efficient vibration attenuation, and good fault-tolerant behavior.

As seen, a scarce number of works can be found in the literature, addressing the use of Neural Networks for vibration suppression in parallel manipulators with flexible links, especially the 3RRR type. In this sense, this work proposes the design of an adaptive neural network control for a flexible-link parallel manipulator. This proposed strategy aims to improve the tracking trajectory accuracy and also attenuate the undesired oscillations caused by the low thickness of the links. The training phase of the ANN is generated based on different trajectory configurations of the prototype. Once tested, additional signals (not used previously in the training and testing phases) are randomly selected to validate the proposed strategy in experimental trials. The control strategy is designed by summing the pose loop, based on a PID strategy, and the adaptive control law, based on the ANN model. This control strategy is then used to to compensate, in real time, the overall dynamics of the flexible-link parallel manipulator, regardless of the trajectory, structural characteristics, model uncertainties, or disturbances. The main contributions of this work are: i) the design of a hybrid adaptive PID + ANN control strategy for trajectory tracking and vibration suppression in flexible-link parallel manipulator, and ii) the experimental evaluation of the Adaptive Neural Network strategy in an instrumented prototype.

The rest of this paper is organized as follows. Section II presents the methodology and instrumentation employed for the work evaluation. Section III details the control development, highlighting the development of both the pose control and active control loops. Section IV presents the results, comparing and improving the proposed control design, while Section V presents the paper's main conclusions.

II. METHODOLOGY

This section presents the parallel manipulator (PM) with flexible links. Figure 1 depicts the required instrumentation used for measuring and controlling the PM. Originally, the prototype was constructed as a 3PRRR mechanism, consisting of three chains with four joints each. In this configuration, the underlined letters correspond to the active joints, while the non-underlined letters represent the passive joints. Moreover, the letters P and R respectively denote the prismatic and revolute joints. However, the prismatic joints are blocked to remove the redundant joints of the original mechanical system, resulting in a 3RRR configuration. Note that the 3RRR configuration consists of three chains with two flexible links each, and three revolute joints, the first of each chain being an active joint. These revolute joints are represented by the motors, as seen by Motor 1, Motor 2, and Motor 3.

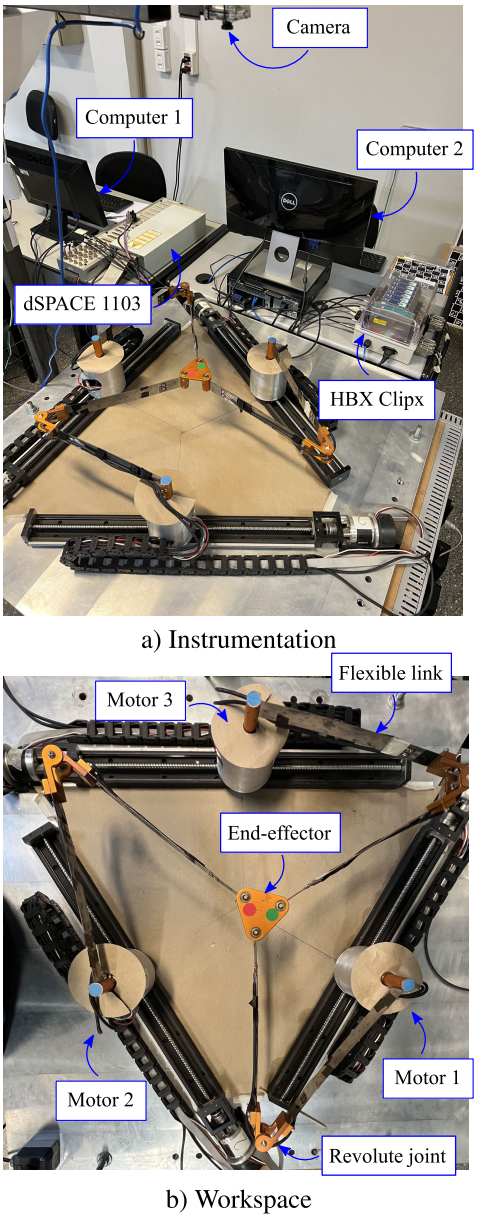


FIGURE 1. Picture of the instrumentation and prototype.

A. INSTRUMENTATION SCHEME

Note that Figure 1 shows the prototype 3RRR configuration and the required instrumentation. The active revolute joints are defined by Motors, which are powered by brushless

Maxon EC60 units. These EC60 motors have a power of 100W and rate of current of 2.3A. Moreover, a nominal rotation of 1200 rpm and nominal torque of 0.82 Nm is achieved, since the motors are coupled to GP52 C planetary reducers with a reduction of 3.5:1. Each motor is equipped with a Maxon EPOS2 50/5 controller, with a power supply of up to 50 Vdc and a current rating of 5 A. Moreover, a control board dSPACE 1103 is used to establish the communication between the equipment and the prototype. The Ethernet is used for the board-computer communication, while CAN protocol establishes the communication between the EPOS controllers and the on-board, including the actuator signals and encoder data. The communication between the EPOS and the dSPACE 1103 has a transmission rate of 250 kbit/s and a sampling time of $T_s = 0.001$ s. The encoders can measure the rotational displacement of each link. Computer 1 is used to design the control strategy by using MATLAB/Simulink. Figure 2 depicts the instrumental scheme of the prototype.

The parallel manipulator is composed of three kinematic chains with two flexible links each. All six flexible links have the same dimensions, including the length of 318.5 mm, height of 3.5 mm, and width of 0.7 mm, and a nominal mass of 7.35g. Note that during the movement of the prototype, a residual deformation can be observed in the experimental responses, mainly due to the material and thickness of the links. Moreover, while the encoders measure only the rotational displacement, the link deformations are acquired by HBM 350-E strain gauges, placed at the endpoint of each link. These signals, collected by the strain gauges, pass through HBM HB40 CLIPX signal amplifiers before being sent to the dSPACE 1103 board. This signal is used to estimate the link deformation over the prototype movement.

In addition, a camera is fixed atop the workspace to capture the real-time images of the prototype pose during its motion. The vision system uses an oCam-5CRO-U camera, equipped with a USB 3.0 interface and a CMOS rolling-shutter image sensor, has a resolution of 640 × 480 pixels, and a frame rate of 60 fps (frames-per-second). The collected images are then processed on Computer 2 using Visual Studio software and the Image Processing Algorithm to estimate the end-effector pose, including its linear displacement and angular orientation.

B. IMAGE PROCESSING ALGORITHM

Two coordinate systems are used to describe the movement of the end-effector from the prototype 3RRR. The fixed reference is defined by the center of the workspace $O(x_c, y_c)$, where x_c and y_c are the planar Cartesian coordinates. On the other hand, the moving system is defined at the center of the red and green markers, as seen in Figure 3. The center of the end-effector, i.e. the pose of the end-effector, is defined by $P(x_p, y_p)$, which allows the Image Processing Algorithm to process each image in real-time and then calculate both the linear and angular displacement of the end-effector. According to the movement of the prototype, a distance is estimated, based on the difference between the fixed and moving coordinate systems, i.e. $x = x_p - x_c$, $y = y_p - y_c$ and $\alpha = \alpha_p - \alpha_c$. This approach enables the estimation of the real characteristics of the prototype, including the rigid kinematics and flexible dynamics.

The Image Processing Algorithm is designed to estimate the pose of the parallel manipulator's end-effector, i.e. its position and angular orientation, in real time using images captured by a fixed camera. The end-effector pose is defined by the corresponding vector $\mathbf{X} = [x, y, \alpha]^T$, where (x, y) correspond to the position of the end-effector, and α to its angular orientation. Figure 3 shows a schematic illustration used for the position-based visual servo control.

Based on Figure 3, there are five circular markers fixed on the end-effector surface, including three circular blue markers fixed to the motors, and two colored markers (red and green) placed on the end-effector surface. While the three blue markers are used to calculate the center of the workspace, based on its geometrical reference, represented by point 0, and the colored markers are used to estimate the linear and angular orientation of the end-effector. The green and red markers are respectively defined by G and R, which are formed by the corresponding vectors (x_G, y_G) and (x_R, y_R). The algorithm processes each image by decomposing it into RGB and HSV color spaces using the OpenCV library, generating six matrices ($\mathbf{R}$, $\mathbf{G}$, $\mathbf{B}$, $\mathbf{H}$, $\mathbf{S}$, $\mathbf{V}$). Filtering operations are then applied to generate a binary matrix ($\mathbf{B}_f$, $\mathbf{G}_f$, $\mathbf{R}_f$), where the number 1 is assigned for the corresponding element, and 0 otherwise. For instance, the elements of matrix $\mathbf{R}_f$ are one if the pixels correspond to red. The filter approach is defined by the following equations $\mathbf{T}_{min} < (\omega_R\mathbf{R} + \omega_G\mathbf{G} + \omega_B\mathbf{B}) < \mathbf{T}_{max}$ and $\mathbf{H}_{min} < \mathbf{H} < \mathbf{H}_{max}$, where ω_j, (j=R,G,B) are the weighting factors of the threshold matrices, while $\mathbf{T}_{min,max} = t_{min,max}\mathbf{1}$ and $\mathbf{H}_{min,max} = h_{min,max}\mathbf{1}$ represents respectively the lower and upper boundaries of the Hue scale matrix (defined by $\mathbf{H}$).

Then, according to the detected marker positions, the end-effector's pose is calculated by the midpoint between the green and red markers defines (x_p, y_p), while the orientation α_p is obtained from the angle of the line connecting them, as seen by

$$\begin{aligned} x_p &= \frac{1}{2}(x_G + x_R), \\ y_p &= \frac{1}{2}(y_G + y_R), \\ \alpha_p &= \arctan\left(\frac{x_R - x_G}{y_R - y_G}\right). \end{aligned} \tag{1}$$

The estimation of the end-effector pose is used as a part of the control input in visual servoing strategies. Further details of the image processing algorithm can be found in [18].

III. CONTROL DESIGN

The control design is accomplished using the full movement of the 3RRR prototype with flexible links. Three control inputs are required to fully control the prototype, which is defined by $\boldsymbol{\tau}_{pose}$. The end-effector pose is estimated using

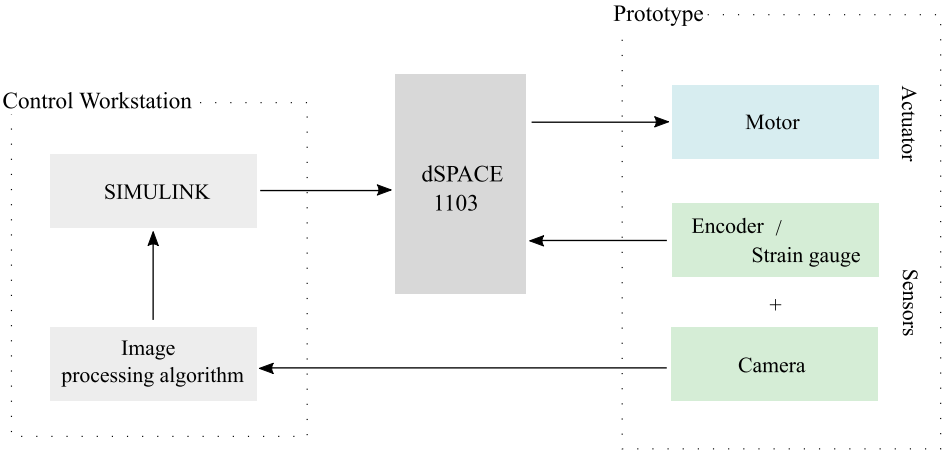


FIGURE 2. Schematic illustration of the prototype communication.

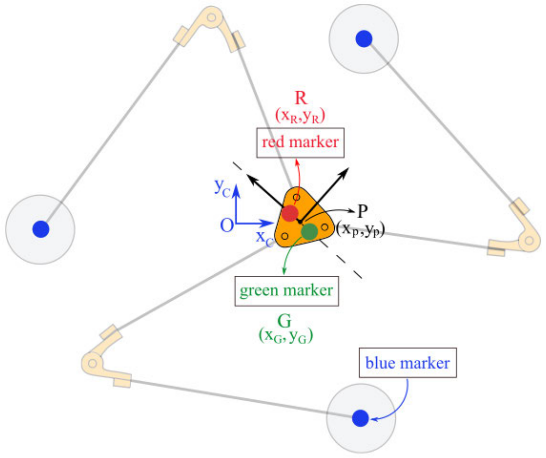


FIGURE 3. Schematic illustration of the 3RRR prototype, showing the position-based visual servo control.

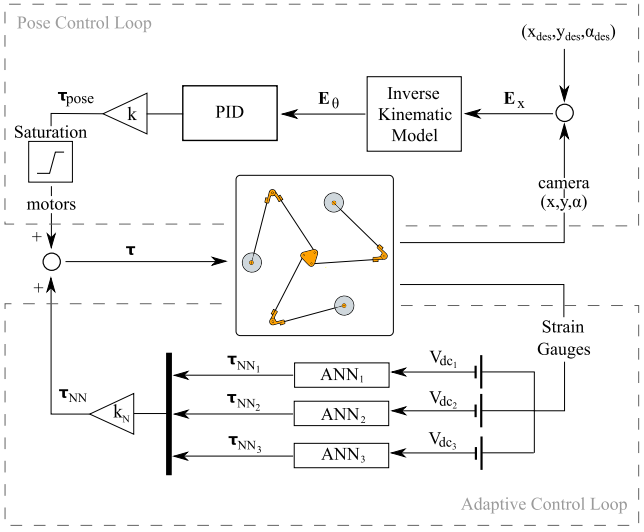


FIGURE 4. Control strategy scheme.

a camera and an Image Processing Algorithm. Then, the pose error ($\mathbf{E}_x$) is calculated based on the difference between the desired signals and those calculated by the camera. The pose error vector $\mathbf{E}_x$ is then transformed into the encoder angle errors $\mathbf{E}_\theta$ through the Inverse Kinematic Model (IKM). Moreover, the PID methodology is used to compute the corresponding control law. Due to the characteristics of the motors, a saturation block is used to limit the current to the motors.

Since there is a high complexity to calculate the mathematical model of the 3RRR with flexible links, a rigid model is assumed, neglecting part of the system's flexible characteristics. In this sense, the strain gauges are used as a sensor to measure the link flexibility over the desired motion. An Artificial Neural Network (ANN) is employed to estimate, in real time, the contribution of flexibility to the overall dynamics, regardless of trajectory, uncertainties, or external disturbances. Figure 4 illustrates the schematic representation of the adaptive control strategy designed in this work.

A. POSE CONTROL LOOP

The Pose Control Loop is designed to estimate the end-effector pose over the desired movement. Note that the Inverse Kinematic Model (IKM) is designed based on the 3 RRR with rigid links, which neglects the effects of the flexibility. Initially, the error is calculated by using the difference of the desired and calculated positions of the end-effector, which is represented by the corresponding vector $\mathbf{E}_x = \mathbf{X}_{des} - \mathbf{X}$, such that $\mathbf{X}_{des} = (x_{des}, y_{des}, \alpha_{des})$ and $\mathbf{X} = (x, y, \alpha)$. Then, an IKM is used to transform the $\mathbf{E}_x$ to $\mathbf{E}_\theta$, to design the control strategy, since all the inputs must be calculated to the encoder reference. The Jacobian matrix establishes the relation between the joint space velocity and task space velocity, which can be defined as $\dot{\mathbf{X}} = \mathbf{J}\dot{\boldsymbol{\theta}}$ or

as $\dot{\boldsymbol{\theta}} = \mathbf{J}^{-1}\dot{\mathbf{X}}$. By expanding the same relationship to the corresponding errors, the relationship between the errors is represented by

$$\mathbf{E}_{\boldsymbol{\theta}} = \mathbf{J}^{-1}\mathbf{E}_{\mathbf{X}} \tag{2}$$

Further details of the kinematic representation of the 3RRR with rigid links can be found in [18]. Regarding the design of pose control, based on the Proportional-Integral-Derivative (PID) control methodology, the control input is designed as a combination of proportional, integral, and derivative terms acting on the pose error. The error in the joint space is defined by $\mathbf{E}_\theta = \boldsymbol{\theta}_{des} - \boldsymbol{\theta}$, while its time derivative can be expressed by $\dot{\mathbf{E}}_\theta = \dot{\boldsymbol{\theta}}_{des} - \dot{\boldsymbol{\theta}}$. While $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ corresponds the actual joint angles, $\boldsymbol{\theta}_{des}$ are corresponding references, as defined by $(\theta_{des_1}, \theta_{des_2}, \theta_{des_3})$. The corresponding PID control law is then expressed as

$$\boldsymbol{\tau}_{pose} = K_p\mathbf{E}_\theta + K_i\int \mathbf{E}_\theta\, dt + K_d\dot{\mathbf{E}}_\theta \tag{3}$$

where K_p, K_i, and K_d are the proportional, integral, and derivative gain matrices, respectively, which are chosen to ensure the desired closed-loop dynamics.

B. ADAPTIVE CONTROL LOOP

Since the flexibility is neglected in the design of the Pose Control Loop, an additional control strategy is included to compensate for the uncertainties dynamics of the prototype. The adaptive control loop employs a neural network approach to estimate the flexible dynamics throughout the desired motion in real time. As illustrated in Figures 1 and 4, a strain gauge is attached to the first link of each chain, enabling the estimation of its flexibility over the desired movement. This loop development is divided into two main phases: training and validation phases. While the training phase comprises the dataset collection and the training of the ANN, validation phase refers to the inclusion of the adaptive loop in the control strategy scheme (see Fig. 4).

1) TRAINING PHASE

In this first stage, a single loop, formed by the Pose Control Loop, is assumed to perform the experimental trials and then collect the data. In addition, the strain gauges are only used as sensor devices to estimate the flexibility of the links during the desired movement. No additional control law, apart from the Pose Control Loop, is included to drive the prototype during the experimental trials. The dataset is created based on random initial locations of the end-effector on the workspace. These multiple trajectories of the prototype enable the assessment of the flexible characteristics over distinct scenarios.

The Artificial Neural Network is formed by multiple hidden layers to map the nonlinear relationships between the input and output variables. Three stages are required to design an ANN: input, hidden, and output. Since each chain presents its own neural network, the input is formed by one element, which is defined by the strain gauge signal of the corresponding link. For instance, Motor 1 has link 1, which represents the strain gauge signal defined by V_{dc_1}. Conversely, the output layer is formed by the corresponding control law, which is calculated previously by employing the Sliding Mode Control methodology. This input signal (strain gauge) is calculated based on SMC methodology to generate the corresponding control law. Initially, a sliding surface is chosen based on the error and its time derivative, as $s = \dot{E}_\theta + \lambda E_\theta$, where E_θ is the error between the desired value (no deflection) 0 and calculated signal from the strain gauge V_{dc_i} of each link. Based on the sliding surface, the corresponding control law can be calculated using the following expression $\tau_{NN_i} = K_N tansig(s_i)$, such that $i = 1, 2, 3$. The tansig function is chosen in this work to attenuate the chattering effects from the SMC methodology. Figure 5 shows some random data from the dataset (inputs and outputs), and Figure 6 illustrates the proposed architecture of an Artificial Neural Network.

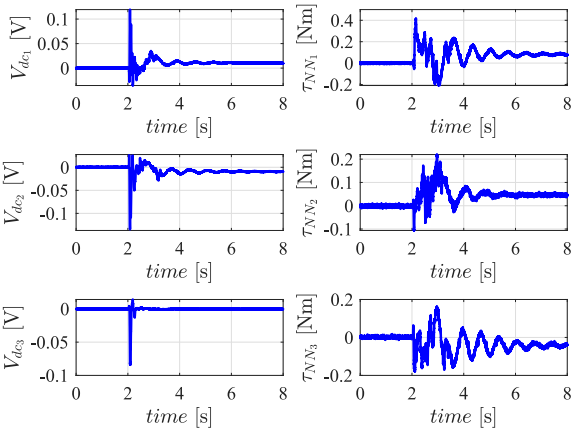


FIGURE 5. Dataset used for training the ANN.

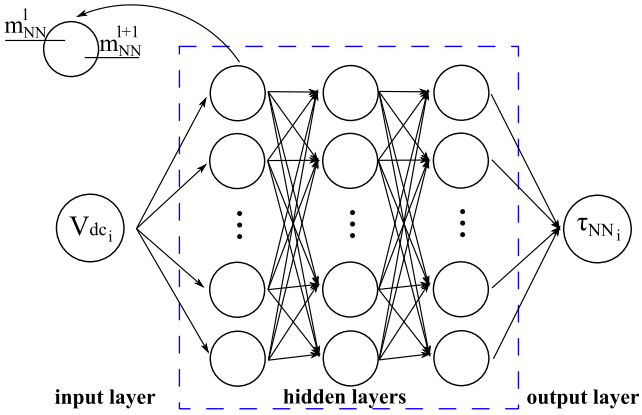


FIGURE 6. Schematic illustration of the Artificial Neural Network.

According to Figure 6, the ANN receives an input (defined by m_{NN}) and propagates through the hidden layers l, calculating their corresponding weight and bias, yielding

$$m^{l+1}_{NN} = W^l m^l_{NN} + b^l \tag{4}$$

where W^l and b^l correspond respectively to the weights and biases at layer l, while m_{NN}^l and m_{NN}^{l+1} are the input and output of each layer. The activation function is applied at each hidden layer to limit the neuron output within a specific range and to enhance the nonlinear characteristics of the model. In this work, the Rectified Linear Unit (ReLU) is adopted as the activation function. During the training phase, the optimization procedure adjusts the weights and biases by minimizing an objective function. Therefore, the loss is quantified based on the Mean Squared Error (MSE) criterion to serve as a performance metric, as seen by

$$\mathscr{L} = \frac{1}{N}\sum_{i=1}^{N}(y_{NN_i} - \hat{y}_{NN_i})^2 \tag{5}$$

where y_{NN_i} and $\hat{y}_{NN_i}$ represent, respectively, the real output and the neural network output values, while N is the length of the vector. The loss function (see Eq. 5) propagates the error from the output layer back to the input layer across all neurons. The partial derivatives $\partial(MSE)/\partial W$ and $\partial(MSE)/\partial b$ are computed with respect to the weights and biases, respectively. These derivatives represent the loss function gradient, which is used for the adjustment of weights and biases during the learning process.

The dataset is partitioned into two main subsets: 90% of the dataset corresponds to the first group, while the remaining 10% is separated for validation. Regarding the first group (90%), two other subsets are defined, allocating 90% for training and 10% for testing. The training procedure is carried out for 4000 epochs, with a neural network architecture consisting of three hidden layers, each comprising 80 neurons.

2) VALIDATION PHASE

The ANN is initially investigated numerically to assess the feasibility and real adaptability of the adaptive control law regardless of the adopted trajectory. Note that the sample used for validation has not been previously used for training nor testing the ANN. Figure 7 shows the validation of the ANN strategy. Despite the differences in the trajectory configuration and deflections of the links, the ANN can be positively adapted according to the dynamics of the system. Therefore, the adaptive control law can be used as a real-time strategy to compensate for the uncertainties and modeling assumptions, regardless of the trajectory, model uncertainties, and structural characteristics.

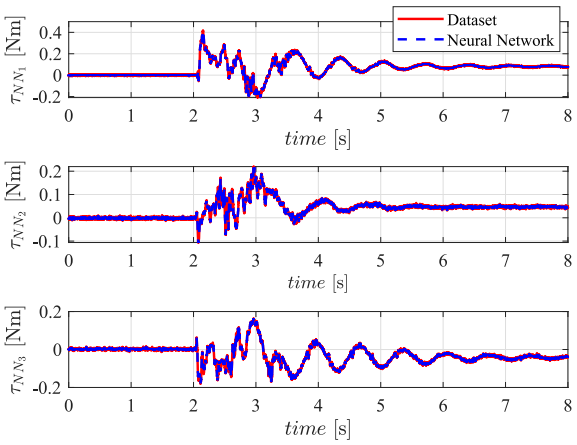


FIGURE 7. Numerical validation of the proposed NN strategy.

Once trained and validated numerically, the adaptive control loop is included in the control strategy scheme for the experimental validation. Then, the corresponding control based on the ANN is included on the control strategy to compensate for the flexible characteristics neglected in the Pose Control Loop development, yielding

$$\boldsymbol{\tau}_{NN} = -K_N \tau_{NN_i} \tag{6}$$

where $\boldsymbol{\tau}_{NN} = [\tau_{NN_1}, \tau_{NN_2}, \tau_{NN_3}]^T$. These adaptive control laws can be adjusted according to the link deflection. Therefore, the final control law can be defined as the sum of both contributions (pose + adaptive), which can adapt in real time based on the movement of the flexible links, as seen

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{pose} + \boldsymbol{\tau}_{NN} \tag{7}$$

The geometrical dimensions of the 3RRR are $\gamma_1 = \frac{\pi}{2}$ rad, $\gamma_2 = \frac{11\pi}{6}$ rad, $\gamma_3 = \frac{7\pi}{6}$ rad, $a = 0.259$m, $h = 0.0285$m and $l_1 = l_2 = 0.3185$m. Note that a corresponds to distance between the workspace center and each motor, h is the distance between the rotational joint of the end-effector and its geometrical center, l_1 and l_2 are respectively the length of links 1 and 2, and γ_i is used to define the angular orientation in relation to the zero point, i.e. relative to the linear joints. The gains are defined as follows $k_p = 900$, $k_i = 300$ and $k_d = 200$, while $K = 50$. The gains of the PID control are experimentally tuned to achieve the best control performance considering the largest workspace.

IV. RESULTS AND DISCUSSION

This section investigates the overall performance of the adaptive control strategy. The proposed strategy is compared with a single pose control to assess the tracking trajectory, estimated by the overshoot and oscillations, over the desired trajectory. Two independent input configurations are chosen to cover the whole workspace: a step input in the x-direction and in the y-direction. While the input in the x-direction is performed from [70 mm, 0 mm, −1.4 rad] to [0 mm, 0 mm, −1.4 rad], the trajectory in the y-direction is generated from [0mm, 70mm, −1.4 rad] to [0 mm, 0 mm, −1.4 rad]. Note that the prototype 3 RRR has coupled dynamics, requiring the actuation of the three motors regardless of the proposed trajectory. As a result, despite considering an input in the x-direction, there is a direct influence on the Cartesian coordinates, including x, y, and α. Ten measurement signals are collected for each trajectory configuration to represent the mean behavior of the Cartesian coordinates. Moreover, a confidence interval of 95% is used to estimate the boundaries of the measurement signal and verify the reproducibility of the results.

In addition, a performance index is used to quantify the overall performance of the proposed strategies, including a single pose loop and the addition of the adaptive loop in the final control strategy. The Integral of Absolute Error (IAE) is widely adopted as a performance index in control theory, since it quantifies the accumulated deviation between the system output and the reference signal over a given time horizon. In this work, the reference signal is defined as zero, since it is assumed to be the final position of the end-effector. From a theoretical perspective, the controller that minimizes the IAE achieves the best nominal performance, as it ensures the smallest possible cumulative tracking error. Note that the system when this index reaches its minimum value. According to [34], the IEA is defined by the following equation $IAE = \int_0^t |c_c(t)|dt$, where t is the final time of the experiment, and $c_c(t)$ is the analyzed coordinate from the Cartesian workspace. In this work, three Cartesian coordinates are investigated x, y, and α.

A. STEP IN THE X-DIRECTION

In this first trajectory configuration, a step in the x-direction is considered. A step of 70mm is applied for the x coordinate, while no real displacement is adopted for the y coordinate. However, due to the coupled dynamics of the 3 RRR, all three motors are required to achieve the desired trajectory. Consequently, the Cartesian coordinates exhibit relative linear and angular displacements. This first analysis evaluates the presence and absence of the adaptive control loop in the final response of the end-effector displacement. Based on Figure 4, the absence of the adaptive control is assumed when $k_N = 0$, while the presence of the adaptive loop in the control design is assumed when $k_N = 50$. Figure 8 shows the linear displacement of the end-effector along the x coordinate for the step input response in the x-direction.

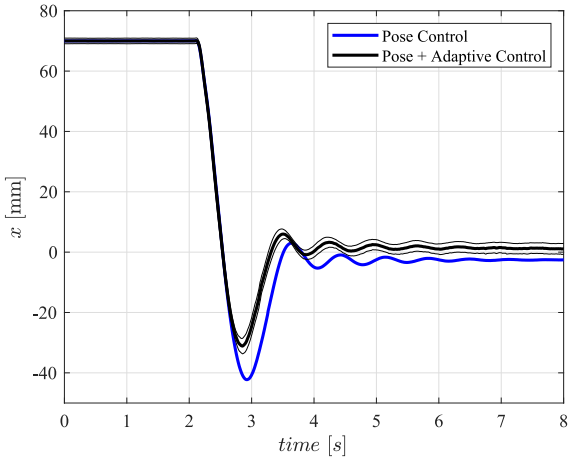


FIGURE 8. End-effector linear displacement (x coordinate) for the step input in the x-direction.

Comparing both control strategies, as seen in Figure 8, a larger overshoot can be noted for the single pose control methodology. For instance, a maximum peak of 42.24 mm is observed in the absence of the adaptive loop, while a maximum peak of 30.54 mm is noted in the presence of the adaptive loop. A percentage index is also investigated to quantify the nominal difference between the two proposed strategies. The percentage index is calculated as $P[\%] = 100(st_{pc} - st_{apc})/st_{pc}$, where st_{pc} and st_{apc} are respectively the corresponding state of the pose control and pose + adaptive control laws. In terms of percentage, a 27.70% reduction is obtained for the maximum peaks. Moreover, not only can the overshoot be significantly attenuated, but the steady-state error and the undesired oscillations are also mitigated with the proposed strategy. Note that the mean signal is closer to the zero nominal value, which corresponds to the desired final position of the end-effector. In addition, by employing a 95% confidence interval analysis, a slight deviation of the mean signal is observed revealing a positive reproducibility.

Besides visual inspection the Integral of Absolute Error performance index is adopted to assess the nominal attenuation by both control strategies. The IAE index is chosen to quantify the combined attenuation of the overshoot, oscillations, and steady-state error, showing a reduction of 10.23% for the x coordinate. Likewise, the y coordinate is investigated when the step input in the x-direction is performed. Since the main motion is generated in the x-direction, more significant oscillations and overshoot are observed in the x coordinate. However, since the prototype exhibits a direct dependence between the Cartesian coordinates, a smaller displacement is also noted in the y-direction. Figure 9 depicts the linear displacement of the end-effector along the y coordinate for the step input response in the x-direction.

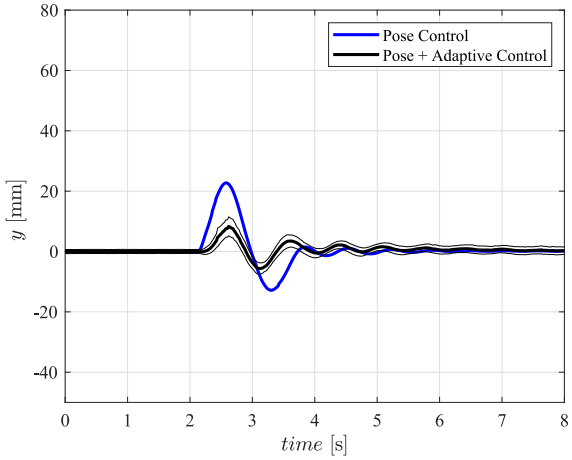


FIGURE 9. End-effector linear displacement (y coordinate) for the step input in the x-direction.

As seen in Figure 9, two main overshoots are observed in this trajectory configuration. When comparing the inclusion of the adaptive loop in the final control strategy, a significant attenuation can also be observed. The two main peaks achieved in the absence of the adaptive control correspond to 22.68 mm and 12.65 mm, while in the presence of adaptive control, the two main peaks are mitigated to 8.10 and 5.60,

respectively. In terms of percentage, the proposed adaptive control strategy results in an average reduction of 60%. Similar to Figure 8, the confidence interval reveals that experimental trial signals exhibit similar behavior, with small deviations from the mean signal. Figure 10 shows the angular rotation of the end-effector for the first trajectory configuration.

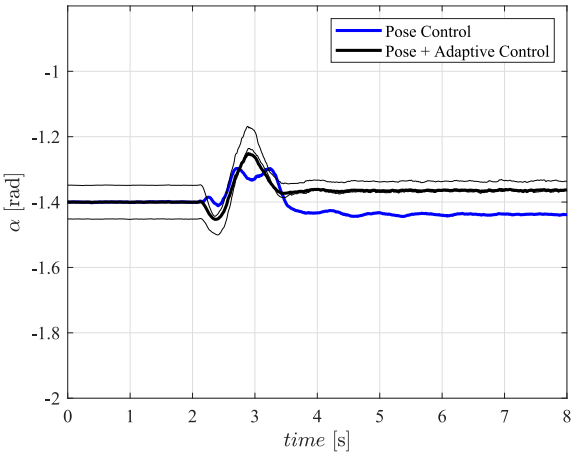


FIGURE 10. Angular rotation of the end-effector for the step input in the x-direction.

Based on Figure 10, no significant differences are observed with the inclusion of the adaptive control law. Aside from a slight increase in the peak, mainly due to the shorter trajectory of the end-effector, which requires a greater rotation, the overall characteristics of α remain similar. Table 1 shows the maximum peaks reached by the end-effector for the step input in the x-direction. Two main peaks are investigated to compare the presence and absence of the adaptive control law.

TABLE 1. Maximum peaks reached by the end-effector for the step in the x-direction.

Control strategy	1^{st} peak		2^{nd} peak	
	x (mm)	y (mm)	x (mm)	y (mm)
Pose Control	42.24	22.68	3.00	12.65
Pose + Adaptive Control	30.54	8.10	3.49	5.60
Percentage P[%]	27.70%	64.29%	+16.33%	55.73%

As depicted in Figures 8 and 9 and Table 1, a significant reduction of the main peaks is achieved in all directions. For instance, the x coordinate exhibits a 27.70% reduction of the main peak, while a 64.29% reduction is observed in the y coordinate. Regarding the second peak, there is a slight increase in the overshoot caused mainly by the mitigation of the steady-state error. Therefore, the inclusion of the adaptive control law enhances the overall performance of the tracking trajectory by showing lower overshoot, undesired oscillations, and steady-state errors. In addition, one of the main advantages of using an adaptive control in the 3RRR prototype is the real time estimation of the flexibility regardless of the adopted trajectory, nominal mass, or deflection of the link. Figure 11 shows the mean value of the strain output of the ANN for the step input in the x-direction. These ANN outputs represent the adaptive control law, as defined by τ_{NN_i}, in which i corresponds to three chains of the prototype.

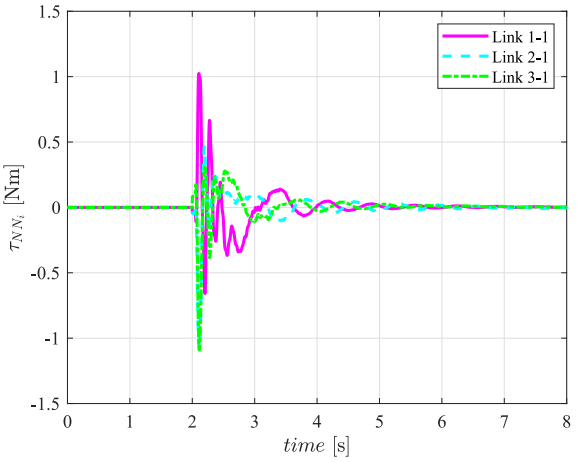


FIGURE 11. Strain output of the ANN for the step input in the x-direction.

B. STEP IN THE Y-DIRECTION

Similar to the first approach, the step input response in the y-direction is employed to investigate the end-effector characteristics in the workspace. In this second trajectory configuration, the path is mainly generated in the other coordinate to cover the whole workspace. The trajectory is generated to remain constant along the x coordinate, while a step input of 70 mm is assumed for the y coordinate. Figure 12 depicts the linear displacement of the end-effector along the x coordinate for the step input in the y-direction. Note that in this case, there is no significant attenuation of the main overshoot. However, a prominent reduction in the steady-state error can be noted.

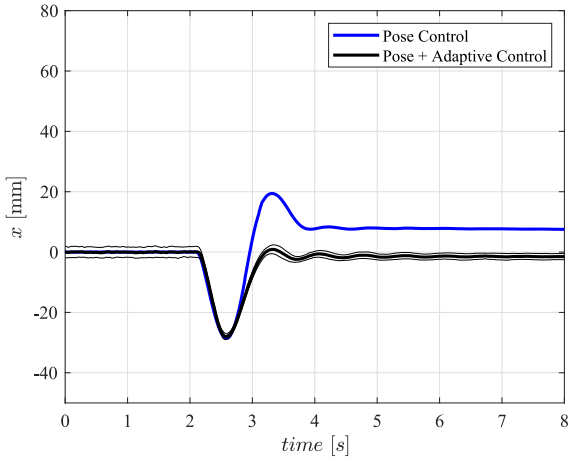


FIGURE 12. End-effector linear displacement (x coordinate) for the step input in the y-direction.

As seen in Figure 12, the second peak exhibits a positive attenuation, and the end-effector final position is achieved closer to zero. For instance, while the first peak has a nominal value of 25.58 mm in the absence of the adaptive control

law, a value of 25.66 mm is observed in the presence of the adaptive law. There is a very slight increase of the main peak, which is compensated by the steady-state error, as seen by the reduction of the second peak from 19.39 mm and 0.83 mm. In terms of percentage, there is an attenuation of 95.72% of the overshoot. Similarly, Figure 13 shows the end-effector displacement along the y coordinate, which is the main direction of the second trajectory configuration.

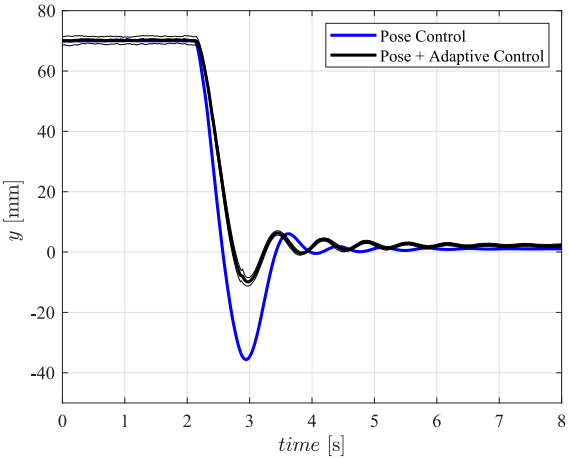


FIGURE 13. End-effector linear displacement (y coordinate) for the step input in the y-direction.

Similarly to the x coordinate, a positive attenuation of the overshoot can be observed. While the two dominant peaks achieved in the absence of the adaptive control correspond to 35.60 and 6.02, the nominal values of 9.87 mm and 6.21 mm are achieved in the presence of the adaptive control law, as seen in Figure 13. This attenuation of the maximum peak represents an average reduction of 72%. Regarding the IAE performance index, an average attenuation of 43.66% and 9.61% is observed for the y coordinate when the overshoot, steady-state, and oscillations are combined. Likewise, a small deviation from the mean signal is observed when the confidence interval is applied, showing that the proposed control exhibits an accurate reproduction of results. Figure 14 shows the angular rotation of the end-effector for the step input in the y-direction. Note that, similar to the first trajectory, no significant difference is noted.

As depicted in Figures 12 and 13 and Table 2, a significant improvement of the tracking trajectory is achieved for most analyzed states. For instance, the y coordinate exhibits an attenuation of 72.28% in the first overshoot, while a slight increase of 3.61% is observed for the second peak. This marginal increase in the second peak is mainly related to the significant decrease in the main overshoot, which demands a little more effort to achieve the final position. Table 3 shows the Integral of Absolute Error performance index calculated for each state. Similar to the overshoot signals, there is a reduction in all directions, ranging from 9.61% to 68.74%. As expected, the main directions of the step input exhibit a more prominent contribution in the IAE performance index.

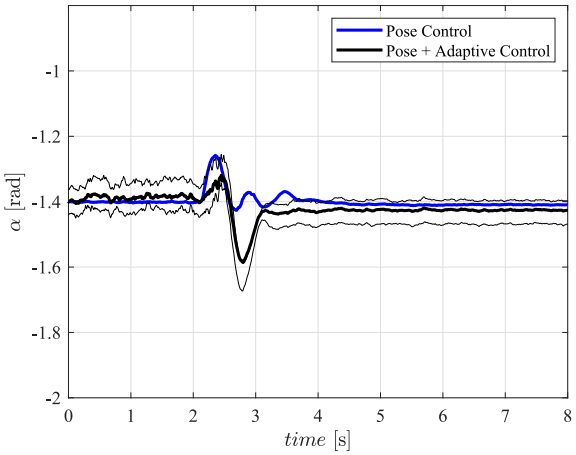


FIGURE 14. Angular rotation of the end-effector for the step input in the y-direction.

TABLE 2. Maximum peaks reached by the end-effector for the step in the y-direction.

Control strategy	1^{st} peak		2^{nd} peak	
	x (mm)	y (mm)	x (mm)	y (mm)
Pose Control	25.58	35.60	19.38	6.02
Pose + Adaptive Control	25.66	9.87	0.83	6.21
Percentage $P[\%]$	+0.31%	72.28%	95.72%	+3.61%

TABLE 3. Integral of absolute error performance index.

Control strategy	Step in the x-direction		Step in the y-direction	
	x (mm)	y (mm)	x (mm)	y (mm)
Pose Control	211.73	18.92	86.20	194.12
Pose + Adaptive Control	190.08	10.66	26.95	175.46
Percentage $P[\%]$	10.23%	43.66%	68.74%	9.61%

An average of 190 nominal values is observed along the main axis, while an average of 20 is noted along the off-axis. The discrepancy in the x coordinate for the step input in the y-direction is observed because of the steady-state error. Therefore, the addition of the adaptive control law into the pose control improves the overall effectiveness of the control strategy by ensuring greater robustness and real-time adaptability, regardless of the desired movement or structural characteristics.

V. FINAL REMARKS

Robotic manipulators are widely used in different industrial applications, including precise welding, the aerospace field, and automatic micro-assembly. Among the most common types of manipulators, those with parallel chains emerge as an interesting alternative. The parallel manipulators with flexible links present several advantages, including lighter weight, higher speed efficiency, and payload-to-weight ratio, since they consume less energy than those with rigid links. Nevertheless, this reduction in mass leads to undesired effects, including the decrease of the tracking trajectory accuracy and the appearance of oscillations. Moreover, these multiple degrees of freedom make the design of an accurate

mathematical model and robust control strategy a challenging task. In this sense, an adaptive control based on an ANN model is designed to cope with environmental changes and model uncertainties in real time. Initially, the ANN model is created based on different trajectories of the parallel manipulator. Then, the ANN model is validated by using random signals and subsequently included to assist in some experimental trials. The control strategy combines the pose loop (based on PID methodology) and the adaptive loop, based on ANN model. Therefore, the proposed control strategy is then used to compensate in real time for the overall dynamics, regardless of trajectory, structural characteristics, model uncertainties, or disturbances.

The adaptive neural network control strategy is compared with a single pose control to assess the tracking trajectory and vibration characteristics throughout the desired trajectory. Two independent input configurations are used to cover the entire workspace: a step input in the x-direction and in the y-direction. Comparing both control strategies, a larger overshoot can be observed for the single pose control loop methodology for all coordinates of the prototype. Experimental results demonstrate that the proposed control approach achieves at least a 27.70% reduction compared to the pose loop scheme, highlighting its effectiveness in enhancing tracking accuracy in flexible-link parallel manipulators. Moreover, the Integral of Absolute Error performance index is adopted to quantify the combined attenuation of overshoot, oscillations, and steady-state error of the coordinates. Similar to the overshoot signals, there is a reduction in all directions, ranging from 9.61% to 68.74%. Therefore, the addition of the adaptive control law to the pose control improves the overall efficiency of the control strategy by ensuring more robustness and adaptability in real time, regardless of the desired movement or structural characteristics.

The recommendations for future works include to perform a different set-up experimental analysis to evaluate the controller performance, including the addition of different masses to the end effector and different trajectory configurations. Moreover, several fault-handling approaches should be addressed to enhance the overall robustness and adaptability of the proposed control strategy.

DECLARATION OF CONFLICTING INTERESTS

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

REFERENCES

[1] M. Sayahkarajy, Z. Mohamed, and A. A. M. Faudzi, "Review of modelling and control of flexible-link manipulators," *Proc. Inst. Mech. Eng. I, J. Syst. Control Eng.*, vol. 230, no. 8, pp. 861–873, 2016.

[2] G. D. Khan, "Adaptive neural network control framework for industrial robot manipulators," *IEEE Access*, vol. 12, pp. 63477–63483, 2024.

[3] Z. Chen, M. Kong, C. Ji, and M. Liu, "An efficient dynamic modelling approach for high-speed planar parallel manipulator with flexible links," *Proc. Inst. Mech. Eng., C, J. Mech. Eng. Sci.*, vol. 229, no. 4, pp. 663–678, Mar. 2015.

[4] R. S. Geronel, G. C. de Oliveira, and M. M. da Silva, "Dual-loop control strategies for vibration attenuation of a flexible parallel manipulator," *J. Vib. Control*, vol. 2024, Dec. 2024, Art. no. 10775463241307702.

[5] X. Zhang, J. K. Mills, and W. L. Cleghorn, "Multi-mode vibration control and position error analysis of parallel manipulator with multiple flexible links," *Trans. Can. Soc. Mech. Eng.*, vol. 34, no. 2, pp. 197–213, Jun. 2010.

[6] D. Liang, Y. Song, Y. Qi, and J. Dong, "Efficient modeling and integrated control for tracking and vibration of a lightweight parallel manipulator including servo motor dynamics," *Mech. Syst. Signal Process.*, vol. 153, pp. 107502–107532, May 2021.

[7] R. S. Geronel, G. C. de Oliveira, G. K. Rodrigues, and M. M. da Silva, "Active vibration control of a flexible parallel manipulator 3RRR," *Robotica*, vol. 43, no. 6, pp. 1–19, Jun. 2025.

[8] L. Jiang, B. Gao, and J. Zhao, "Kinematic and static analysis of a cable-driven parallel robot with a flexible link spine," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2015, pp. 31–36.

[9] Z. Chen and X. Wang, "Dynamic modeling and residual vibration suppression of the redundantly-actuated cable driving parallel manipulator," *IEEE Access*, vol. 8, pp. 99422–99430, 2020.

[10] D. Subedi, I. Tyapin, and G. Hovland, "Dynamic modeling of planar multi-link flexible manipulators," *Robotics*, vol. 10, no. 2, pp. 70–85, May 2021.

[11] Z. Jin, Y. Zhao, Y. Sun, and Y. Liu, "Force feedback controller of a parallel haptic device via online adaptive dynamic programming," *Mechatronics*, vol. 106, pp. 103293–103305, Apr. 2025.

[12] S. M. Mahdi, A. I. Abdulkareem, A. J. Humaidi, A. K. Al Mhdawi, and H. Al-Raweshidy, "Comprehensive review of control techniques for various mechanisms of parallel robots," *IEEE Access*, vol. 13, pp. 63381–63416, 2025.

[13] F. S. Lee and C. I. Lin, "Controller design for a delta robot using Lagrangian multipliers," *Res. Square*, pp. 1–13, 2021.

[14] J. F. García-Samartín and A. Barrientos, "Kinematic modelling of a 3RRR planar parallel robot using genetic algorithms and neural networks," *Machines*, vol. 11, no. 10, pp. 952–966, Oct. 2023.

[15] Q. Huang, G. He, G. Feng, and B. Ding, "A novel cooperative control strategy for a three-degree-of-freedom pneumatic parallel mechanism," *Actuators*, vol. 13, no. 3, pp. 89–102, 2024.

[16] P. Bengoa, A. Zubizarreta, I. Cabanes, A. Mancisidor, C. Pinto, and S. Mata, "Virtual sensor for kinematic estimation of flexible links in parallel robots," *Sensors*, vol. 17, no. 9, pp. 1934–1948, Aug. 2017.

[17] Z. Zake, F. Chaumette, N. Pedemonte, and S. Caro, "Vision-based control and stability analysis of a cable-driven parallel robot," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1029–1036, Apr. 2019.

[18] F. T. Colombo and M. M. da Silva, "Two hybrid model-based control strategies for a flexible parallel planar manipulator," *Control Eng. Pract.*, vol. 127, pp. 1–9, Oct. 2022.

[19] L. Zhang, G. Song, C. Yang, C. Zou, H. Cheng, R. Huang, J. Qiu, and Z. Yin, "A parallel compliant leg for energy-efficient walking of exoskeleton–Walker systems," *Mechatronics*, vol. 98, pp. 103110–103125, Sep. 2024.

[20] M. Cheng, X. Zhang, Q. Ding, J. Zhang, and B. Xu, "Residual vibration suppression of a large-size flexible hydraulic manipulator under external disturbance with accurate positioning," *Mechatronics*, vol. 110, pp. 103355–103370, Oct. 2025.

[21] H. Yang, Y. Jiang, and S. Yin, "Adaptive fuzzy fault-tolerant control for Markov jump systems with additive and multiplicative actuator faults," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 4, pp. 772–785, Apr. 2021.

[22] Z. Zhang, Q. Meng, Z. Cui, M. Yao, Z. Shao, and B. Tao, "Machine learning applications in parallel robots: A brief review," *Machines*, vol. 13, no. 7, pp. 565–582, Jun. 2025.

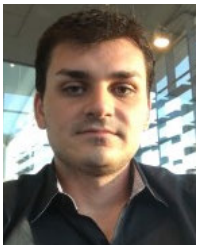
[23] M. Adel, S. M. Ahmed, and M. Fanni, "End-effector position estimation and control of a flexible interconnected industrial manipulator using machine learning," *IEEE Access*, vol. 10, pp. 30465–30483, 2022.

[24] D.-Y. Yu, "Parallel robots pose accuracy compensation using artificial neural networks," in *Proc. IEEE Int. Conf. Mechatronics Autom.*, Aug. 2008, pp. 750–754.

[25] D. Yu, "A new pose accuracy compensation method for parallel manipulators based on hybrid artificial neural network," *Neural Comput. Appl.*, vol. 33, no. 3, pp. 909–923, Feb. 2021.

[26] Y. Li and Y. Wang, "Trajectory tracking control of a redundantly actuated parallel robot using diagonal recurrent neural network," in *Proc. 5th Int. Conf. Natural Comput.*, vol. 2, Aug. 2009, pp. 292–296.

[27] Z. Liu, K. Peng, L. Han, and S. Guan, "Modeling and control of robotic manipulators based on artificial neural networks: A review," *Iranian J. Sci. Technol., Trans. Mech. Eng.*, vol. 47, no. 4, pp. 1307–1347, Dec. 2023.
[28] B. Rahmani and M. Belkheiri, "Adaptive neural network output feedback control for flexible multi-link robotic manipulators," *Int. J. Control*, vol. 92, no. 10, pp. 2324–2338, Oct. 2019.
[29] V.-T. Nguyen, S.-F. Su, N. Wang, and W. Sun, "Adaptive finite-time neural network control for redundant parallel manipulators," *Asian J. Control*, vol. 22, no. 6, pp. 2534–2542, Nov. 2020.
[30] Q. Zhang, X. Zhao, L. Liu, and T. Dai, "Adaptive sliding mode neural network control and flexible vibration suppression of a flexible spatial parallel robot," *Electronics*, vol. 10, no. 2, pp. 212–225, Jan. 2021.
[31] L.-H. Yu, Z.-C. Qiu, and X.-M. Zhang, "Radial basis function neural network vibration control of a flexible planar parallel manipulator based on acceleration feedback," *J. Vib. Control*, vol. 28, nos. 3–4, pp. 351–363, Feb. 2022.
[32] Y. Liu, X. Yao, and W. Zhao, "Distributed neural-based fault-tolerant control of multiple flexible manipulators with input saturations," *Automatica*, vol. 156, Oct. 2023, Art. no. 111202.
[33] L. Li and J. Liu, "Neural-network-based adaptive fault-tolerant vibration control of single-link flexible manipulator," *Trans. Inst. Meas. Control*, vol. 42, no. 3, pp. 430–438, Feb. 2020.
[34] R. C. Dorf and R. H. Bishop, *Modern Control Systems*. New York, NY, USA: Pearson, 2011.



RENAN SANCHES GERONEL received the B.Sc. and M.Eng. degrees in mechanical engineering from São Paulo State University, Ilha Solteira, Brazil, in 2016 and 2018, respectively, and the Ph.D. degree in mechanical engineering from the Unesp in collaboration with École de Technologie Supérieure, Montreal, Canada, in 2023. Currently, he is a Postdoctoral Researcher with the University of São Paulo, Brazil. His research interests include modeling and control of nonlinear systems, smart materials, robotic systems, and UAV-based systems.

FÁBIO LÚCIO FELIX received the B.Sc. degree in mechanical engineering from the Federal University of Triângulo Mineiro, Uberaba, Brazil, in 2018, and the M.Sc. degree in mechanical engineering from the University of São Paulo, Brazil, where he is currently pursuing the Ph.D. degree in mechanical engineering. His research interests include kinematic and dynamic modeling of robotic systems and artificial intelligence methodologies.

MAÍRA MARTINS DA SILVA received the B.Sc. and M.Sc. degrees in mechanical engineering from the University of São Paulo, Brazil, in 2002 and 2004, respectively, and the Ph.D. degree from Katholieke Universiteit Leuven, Belgium, in 2009. Since 2010, she has been a Professor with São Carlos School of Engineering, University of São Paulo. She coordinates a research project on design and control of flexible manipulators. Her research interests include motion and vibration control, optimization, kinematic, and dynamic modeling of robotic systems.

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - ROR identifier: 00x0ma614