End-Effector Position Estimation and Control of a Flexible Interconnected Industrial Manipulator Using Machine Learning

MUHAMMAD ADEL¹, SABAH M. AHMED¹,², AND MOHAMED FANNI.¹,³
¹Mechatronics and Robotics Eng. Dept., Egypt-Japan University of Science and Technology, Alexandria, Egypt
²On leave: Electrical Engineering Department, Faculty of Engineering, Assiut University, Egypt
³On leave: Production Engineering Dept., Faculty of Engineering, Mansoura University, Mansoura, Egypt
Corresponding author: Muhammad Adel - muhammad.adel@ejust.edu.eg

ABSTRACT
The control of flexible robot manipulators is a challenging task, especially when one considers parallel and interconnected manipulators under flexibility considerations. This paper proposes a method to estimate the position of the end-effector of a flexible interconnected manipulator based on a virtual sensor principle and function approximation schemes. By using SolidWorks/MSC ADAMS software, we developed a virtual prototype of a flexible interconnected manipulator, and rigorously evaluated the feasibility of using function approximation schemes such as Neural Networks (NN), Support Vector Machines (SVM), and Gaussian Process (GP) in estimating the deflection error arising due to the flexibility of the robot structure. Our rigorous computational experiments have shown that: (1) the NN, SVM, and GP models were able to attain the promising and reasonable prediction accuracy; (2) a feedforward NN with 535 neurons and an ascending distribution of its nodes achieves the best prediction and generalization to unseen environments (the upper bound of the error was $0.15 \times 10^{-3}$ m); implying the robust estimation of the position of the end-effector under flexibility considerations, and (3) the control based on the inverse Jacobian and a NN-based estimator was able to follow a sinusoidal trajectory with reasonable tracking and error performance in MSC ADAMS & MATLAB/Simulink co-simulation. Our results show the feasibility and effectiveness of the nonlinear relationships learned by NN, SVM, and GP in aiding estimation and control of the position of the end-effector of the flexible manipulator with a promising/desirable capability.

INDEX TERMS Neural Networks, MSC ADAMS, Deflection Estimation, Flexible Robots, Gaussian Process, Interconnected Manipulator, Machine learning, Support Vector Machines.

I. INTRODUCTION

RECENTLY, robot manipulators are being used in a plural number of industrial applications, such as pick and place operations, welding, spray-painting, cleaning, operations in hazardous environments, maintenance, assembly, and machining processes [1]. The great interest in the above applications has its source in combining minimal cost with adaptability and quality with productivity.

The industrial robots based on serial configurations have attained the major interest in industrial stakeholders, mainly due to the large workspace-to-size ratio [2]. However, the conventional serial robot manipulators require high weights of bulky material in order to achieve the high stiffness and the high positional accuracy in tracking. However, the increase in materials and weights not only limits the operating speed of the manipulator but also demands larger and more powerful actuators [3]. Parallel manipulators overcome the problems and drawbacks of the manipulators based on serial configuration. Basically, parallel manipulators are preferred in high-speed motion applications due to their advantages in higher load/weight ratio, high stiffness, high precision, and high-speed [4], [5]. Nevertheless, parallel manipulators still suffer from a low workspace, and abundant singularities within the workspace [6].
In line with the above observations, measuring the deformation of robot manipulators with flexible links has attracted the attention of the community. The flexible robotic systems are continuous dynamical systems with infinite number of degrees of freedom (DOF). Since these dynamical systems are governed by non-linear, coupled, ordinary, and partial differential equations, the mathematical model is normally not feasible [7]. In order to discretize these continuous systems, Assumed Mode Method (AMM), which has been applied in [8]–[10], and Finite Element Method (FEM), which has been applied in [11], have been used [7]. Commonly, researchers utilize the Euler-Bernoulli beam theory as in [12] which extended and developed from Timoshenko beam theory that has later been used and developed as in [13], [14] to include the effects of shear and rotational inertia. Since driving these equations of motion for the robotic systems that have more than one joint is a time consuming, recursive algorithms have been derived such in [15], [16]. However, these methods are still complicated and needs a lot of time and effort, especially when working with complicated structures.

Different approaches have attracted the attention of the research community, and the sensor-based approach is a well-known approach [17]. One of the sensor-based approaches is the strain gauge, which is easy-to-use and cost-effective [18]. However, the performance of strain gauges is subject to conditions and variations of temperature, noise, and electromagnetic interference [19]. An accelerometer is another type of sensor that provides an accurate vibration feedback rate for flexible manipulators [20]. However, using accelerometers require a computationally expensive approach to minimize the accumulation of errors. More recently, there exists the trend towards using multimodal techniques in computer vision [21], [22]. However, vision-based approaches are limited by the view range, are prone to obstacle/ manipulator obstruction, and are bounded by computational resources [4].

On the other hand, interconnected manipulators, which offer the advantages of both serial and parallel robots [23], [24] often have actuators/motors at/near the fixed platform, so lightweight links can be used to achieve higher speed and acceleration in addition to low consuming power [25]. But a large number of passive joints degenerates the rigidity of interconnected manipulators, which leads to positional error at the end-effector due to the flexibility considerations of the links. For stiff or rigid manipulators, the determination of the end-effector location can be realized easily using the active links. For stiff or rigid manipulators, the determination of the position of the end-effector consider not only the position of the active joints but also the deformation of its flexible links [4]. Therefore, the stiff element assumption for flexible manipulators is insufficient to capture the location of the end-effector accurately [26], [27].

This paper proposes the idea of using the virtual sensor theory in conjunction with function approximation schemes to estimate the position of the end-effector of a flexible interconnected manipulator. The basic idea of our approach is to build function approximation schemes that estimate the deflection error of the position of the end-effector as a function of the values of the angular deflection of the passive joint angles. As such, the function approximation schemes can be used not only to predict arbitrary manipulator trajectories under flexibility considerations of its structure but also to design control approaches that consider the deflection error explicitly. Concretely speaking, our contributions are as follows:

- We developed a virtual prototype of an interconnected manipulator in SolidWorks/MSC ADAMS software where links are treated as flexible structures. Since the manipulator has a complex structure, it is unrealistic to find the closed-form mathematical representations able to describe the position of the end-effector under flexibility considerations.
- We evaluated the feasibility of using function approximation schemes derived from the machine learning community such as NN, SVM, and GP to estimate the deflection error at the end-effector of an interconnected manipulator based on the readings of the encoders located at passive joints. We also evaluated the generalization abilities of NN, SVM, and GP in estimating the trajectory of the flexible interconnected manipulator in environments that are relevant to real-world applications, that is, trajectories comprising a pick and place task, a spiral, a lemniscate, a curved trajectory, and an ellipsoid. Our rigorous computational experiments have shown that: (1) the NN, SVM, and GP models can attain reasonable prediction accuracy in estimating the position of the end-effector of the interconnected manipulator, and (2) a feedforward NN with 535 neurons with an ascending distribution of its nodes in 6 hidden layers achieves the best prediction and generalization to unseen environments. This indicates the enhanced and robust prediction of the position of the end-effector under flexibility considerations (the upper bound of the error was $0.15 \times 10^{-3}$ m).
- We evaluated the applicability of using function approximation in developing control schemes under flexibility considerations in a co-simulation between MATLAB-Simulink and MSC ADAMS software. This shows that: (1) the conventional joint space control technique was unable to reach the desired target due to its inability to consider the deflection errors in the flexible manipulator, and (2) the task space control based on the inverse Jacobian and a NN-based estimator was able to follow a sinusoidal trajectory with a very small error that does not exceed $1 \times 10^{-3}$ m in the three axes.

Our results show that it is possible to use the nonlinear relationships learned by NN, SVM, and GP to aid in the prediction and the control of the position of the end-effector of the flexible manipulator with promising/desirable accuracy. This paper is organized as follows: the proposed approach
is presented in Section II which describes the structure of the interconnected manipulator, the Forward Kinematics, Inverse Kinematics, the Forward Velocity Kinematic Analysis the virtual sensor theory and the function approximation schemes (ANN, SVM, and GP). Then, the simulation experiments are discussed in Section III illustrating the configuration of the flexible interconnected manipulator, discussing the prediction performance of the machine learning techniques (ANN, SVM, and GP), and describing the model testing. The last subsection D discusses the application in control design using the joint space control and the task space control techniques. Finally, the conclusion and future research directions are given in Section IV.

II. PROPOSED APPROACH

In this section, we describe the basic concepts behind the structure of the flexible interconnected manipulator and the basic principles behind the introduction of the function approximation schemes aiding to estimate the position of the end-effector of the flexible manipulator.

A. INTERCONNECTED MANIPULATOR

In this paper, we consider the interconnected manipulator as shown in Fig. 1-(a), which is basically a hybrid between serial and parallel manipulators. As shown in Fig. 1-(b), the manipulator consists of several links, each of which is composed of alloy steel with Young’s modulus of 210 GPa, Poisson’s ratio of 0.28, and density of 7700 kg/m³ [28]. The links labeled with numbers 1, 2, 3, 4, 6, 8, 11, 12, 19, and 21 are considered the main links that construct the serial chains, among which:

- link 1 has a length of 0.162 m, and other links have a length of 0.444 m,
- links 11 and 12 have cross-section 0.020 m thickness and a width of 0.110 m, and other links have cross-section size 0.018 m thickness and a width of 0.050 m,
- link 4 contains three Passive revolute joints, and the difference between the first two is 0.162 m while the distance between the other two joints is 0.444 m.

Also, Fig. 1-(b)-(e) shows that the manipulator uses seven parallelograms to move the end-effector in three perpendicular directions with a fixed orientation.

- Five of these parallelograms are in a parallel plane, in which three out of the five parallelograms facilitate the movement of the end-effector in the longitudinal and vertical directions without changing its orientation in the vertical plane as shown in Fig. 1-(c)-(d), and the other two of the five parallelograms have the role of eliminating the internal singularities in the workspace.
- The remaining two parallelograms out of the seven, as shown in Fig. 1(d), are orthogonal to the aforementioned five parallelograms. These two parallelograms, along with the above-mentioned first three parallelograms, facilitate the movement of the end-effector in a lateral direction without changing its orientation in the horizontal plane [29].

Since the interconnected manipulator shown by Fig. 1 is inspired by a cantilever-like structure with rotary actuators at/near the base, it brings the advantages of both parallel and serial manipulators. Thus, the hybrid manipulator shown by Fig. 1 can achieve a higher load/weight ratio, stiffness, precision, and speed comparable to those of parallel robots. At the same time, it has a large workspace volume comparable to those of serial robots and higher than those of its peers such as the 3D pantograph [30], [31], and Pantopteron [32]. As seen on the schematic diagram of the proposed model shown in Fig. 1-(e), point F reaches the boundary of the workspace at/near the base, it brings the advantages of both parallel and serial robots. At the same time, it has a large workspace volume comparable to those of serial robots and higher than those of its peers such as the 3D pantograph [30], [31], and Pantopteron [32].

B. FORWARD KINEMATICS (FK)

The FK relates the end-effector’s position with the state of the active joints. By observing Fig. 1-(e), the manipulator consists of three active joints $\theta_1$, $\theta_2$, and $\phi$, thus the generalized joint position of the manipulator can be expressed by the vector $q = (\theta_1, \theta_2, \phi)$. Therefore, to determine the end-effector’s position $p^r = (X_r, Y_r, Z_r)$ with respect to the fixed reference frame $\alpha$-xyz, the manipulator closure loop equation is obtained as follows:

\[
\overrightarrow{BF} = \overrightarrow{BE} + \overrightarrow{EF},
\]

where, $\overrightarrow{BF}$ is the end-effector’s position vector, and the end-effector’s position of the manipulator can be obtained in the Cartesian space as follows:

\[
X_r = b(\cos \theta_1 - \cos \theta_2) \cos \phi \tag{3}
\]
\[
Y_r = b(\sin \theta_1 - \sin \theta_2) \tag{4}
\]
\[
Z_r = b \sin \phi(\cos \theta_2 - \cos \theta_1) \tag{5}
\]

In the above, the subscript $r$ denotes the implicit notion of rigid structures.

C. INVERSE KINEMATICS

The analysis of the inverse kinematic for the interconnected manipulator concerns with the determination of the joint angles $(\theta_1, \theta_2, \phi)$ corresponding to given desired position of the end-effector. The inverse kinematics can be obtained from Eqs. (3)-(5). Due to the non-linearity existed in these equations, there are 4 solutions for each angle, the symbolic toolbox in MATLAB has been used to find these solutions. We choose the correct solution that corresponds to the manipulator’s configuration as Eqs. (9)-(11) present below:

\[
Q_1 = X_r^2 + Y_r^2 + Z_r^2 \tag{6}
\]
$Q_2 = -Y_r^2 + Q_1$ \hspace{1cm}(7) \\
$Q_3 = -Q_1 Q_2 (-4b^2 + Q_1)$ \hspace{1cm}(8) \\
$\theta_1 = \text{atan2} \left( \frac{\sqrt{Q_3 + Q_1 Y_r}}{bQ_1}, \frac{\sqrt{Q_3 Y_r + Q_1 Q_2}}{\sqrt{1/Q_2} bQ_1 Q_2} \right)$ \hspace{1cm}(9) \\
$\theta_2 = \text{atan2} \left( \frac{\sqrt{Q_3 - Q_1 Y_r}}{2bQ_1}, -\frac{\sqrt{Q_3 Y_r + Q_1 Q_2}}{2 \sqrt{1/Q_2} bQ_1 Q_2} \right)$ \hspace{1cm}(10) \\
$\phi = \text{atan2} \left( -Z_r \sqrt{1/Q_2}, X_r \sqrt{1/Q_2} \right)$ \hspace{1cm}(11) \\

$D. $ **VELOCITY KINEMATIC ANALYSIS**

By taking the derivative of the equations of FK, the forward velocity kinematics are as follows:

\[
\begin{bmatrix}
    \dot{X}_r \\
    \dot{Y}_r \\
    \dot{Z}_r
\end{bmatrix} = J \begin{bmatrix}
    \dot{\theta}_1 \\
    \dot{\theta}_2 \\
    \dot{\phi}
\end{bmatrix} \tag{12}
\]

\[
J = \begin{bmatrix}
    -bc\phi s_1 & b_\phi (c_1 - c_2) & bc_\phi s_2 \\
    -bs_\phi s_1 & -bc_\phi (c_1 - c_2) & -bs_\phi s_2 \\
    -bc_1 & 0 & -bc_2
\end{bmatrix} \tag{13}
\]

where, $s_1 = \sin \theta_1$, $s_2 = \sin \theta_2$, $s_\phi = \sin \phi$, $c_1 = \cos \theta_1$, $c_2 = \cos \theta_2$, $c_\phi = \cos \phi$, and $J$ stands for the Jacobian matrix.

The inverse velocity kinematic analysis can be obtained by taking the inverse of the Jacobian matrix presented Eq. 13.
E. VIRTUAL SENSOR THEORY

The relationships encompassing the kinematics of the aforementioned interconnected manipulator permit to calculation of the position of the end-effector as a function of the angles of the three active joints $\theta_1, \theta_2,$ and $\phi$. However, the conventional kinematic relationships are unable to consider deformations in the overall structure, which is a byproduct when links are flexible and subject to deformations, thus rendering end-effector positions which are different from the ones predicted by Eq. 3 - Eq. 5 [4], [33]. To account for deformations in the structure, the kinematic equations require to compute the end-effector position in terms of not only the rigid joint properties, but also the flexible features of the manipulator, such as the transverse deflection $q_{fd}$ and the slope deflection $q_{fs}$ of the flexible links as shown in Fig. 2. Here, the slope deflection $q_{fs}$ affects the angular rotation of the passive joints in parallel or interconnected manipulators. In this line, it is shown in [4], that measuring the error of angular rotation of a passive joint by an encoder leads to the estimation of the slope deflection. The transverse deflection $q_{fd}$ can be obtained with reasonable accuracy by using the normal model analysis [4]. Therefore, the general motion of any system is composed as a superposition of its normal modes. And, if a particular mode has been considered, and the modal motion of a single DOF has been measured, the complete set of DOF modal motions can be estimated from (14).

$$q_{dof} = \sum_{k=1}^{n_{dof}} X_k \epsilon_k, \quad (14)$$

where, $X_k$ is the $k^{th}$ eigenvector or mode shape and $\epsilon$ is the $k^{th}$ modal motion [4]. Due to the limited band width of the working frequencies of the actuator system, (14) can be computed by (15).

$$q_{dof} = \sum_{k=1}^{n_{dof}} X_k \epsilon_k \approx X_1 \epsilon_1 \quad (15)$$

So that, the Finite Element Method and Euler-Bernoulli Beam theory can be used to model the flexible link. Then, a high-resolution optical encoder can be used to measure the angular deformation so that by substituting into (2), the deflection at the tip of the link can be easily estimated. In such a way, it is possible to compute the actual position of the end-effector of the manipulator.

The above-described approach is useful when the number of links is small or the structure of the manipulator is relatively simple, such as the case of Delta or other parallel manipulators. Since the considered interconnected manipulator is relatively more complex compared to the Delta manipulator in [4], the above-described methodology cannot be used straightforwardly. However, the aforementioned concept provides with important information which is crucial to our proposed approach. That is, measuring the error of the angular rotation of the passive joint angles can lead to the estimation of the actual position of the end-effector.

F. FUNCTION APPROXIMATION SCHEMES

Due to the relatively complex model of the interconnected manipulator, it is unfeasible to obtain an accurate mathematical model being able to describe the position of the end-effector under flexibility considerations. Thus, in this paper, we explore the feasibility of using machine learning methods aiding to build forward relationships, as shown by Fig. 3, derived from the principle of the virtual sensor theory, which has the potential benefits:

- to install high-resolution optical encoders at the passive joints of the main parallelogram as shown in Fig 3,
- to capture the dynamics of the complex interconnected manipulator system.

By observing Fig. 3, it is possible to let a function approximation scheme (FA) to estimate/predict the error (deflection) of the position of the end-effector occurring due to the flexible links in the structure. As a result, one can use the FK and the estimated deflections to calculate the actual position of the end-effector under flexibility considerations. In this figure, the active joint angles $q = (\theta_1, \theta_2, \phi)$ are used to estimate the end-effector position $p^* = (X_r, Y_r, Z_r)$ from the FK, as well as the deflection angles of the passive joints. The angular deflection $q_a = (q_{a1}, q_{a2}, q_{a3}, q_{a4})$, can be obtained by subtracting these rigid passive joint angles from the corresponding flexible passive ones obtained from MSC ADAMS software. In practices, these latter angles are measured by optical encoders. Then, it is possible to estimate the deflection error $e = (e_x, e_y, e_z)$ from the mapping:

$$f : q_a \rightarrow e, \quad (16)$$

where, $f$ denotes the function approximation scheme. Therefore, from Fig. 3, one can determine the end-effector’s position with $p = p^* + e$, which considers the estimations from the (FK) as well as the estimation of the deflection error due to flexible structure. For simplicity and without loss of generality, we study the performance/feasibility of a well-known subset of function approximation schemes derived from the machine learning community. In this section, such frameworks are briefly described.

1) Artificial Neural network (ANN)

An ANN has the ability to learn and model complicated relationships between inputs and outputs [34]. Due to their ability to learn from examples, the ANN has become a flexible and powerful tool to solve regression and classifications problems in many fields [35]. Basically, the ANN models interrelated
layers of artificial neurons (nodes), organized hierarchically using an input layer, one or more hidden layers, and an output layer [36].

Since we study the feasibility of estimating the deflection errors as a function of the angular deflection of the passive joint angles, our target is then to learn mapping functions of the form:

\[ f_i : q_a \rightarrow e \]  \hspace{1cm} (17)

where, \( f_i \) denotes the \( i \)-th regressor model using \( q_a \) as an input. Here, due to the fact of having four passive joints \((q_{a_1}, q_{a_2}, q_{a_3}, q_{a_4})\), four digital encoders will be necessary to measure the angular deflection, each located near a passive joint. In order to study the contribution on the prediction performance of measuring the angular deflection in each passive joint, we evaluated all possible combinations of the below-described input modes to build function approximations, thus \( i = 1, 2, ..., 15 \), and

\[
\begin{align*}
q_a^1 &= q_{a_1} \\
q_a^2 &= q_{a_2} \\
q_a^3 &= q_{a_3} \\
q_a^4 &= q_{a_4} \\
q_a^5 &= (q_{a_1}, q_{a_2}) \\
q_a^6 &= (q_{a_1}, q_{a_3}) \\
q_a^7 &= (q_{a_1}, q_{a_4}) \\
q_a^8 &= (q_{a_2}, q_{a_3}) \\
q_a^9 &= (q_{a_2}, q_{a_4}) \\
q_a^{10} &= (q_{a_3}, q_{a_4}) \\
q_a^{11} &= (q_{a_1}, q_{a_2}, q_{a_3}) \\
q_a^{12} &= (q_{a_1}, q_{a_2}, q_{a_4}) \\
q_a^{13} &= (q_{a_1}, q_{a_3}, q_{a_4}) \\
q_a^{14} &= (q_{a_2}, q_{a_3}, q_{a_4}) \\
q_a^{15} &= (q_{a_1}, q_{a_2}, q_{a_3}, q_{a_4})
\end{align*}
\]

2) Support Vector Machine (SVM)

Introduced initially as a pattern recognition tool by Vapnik and colleagues at AT&T Bell laboratories [37], SVM has been extended to solve regression problems by constructing multidimensional splines and solving linear equations [38], [39], thus the main idea behind SVM is to create hyperplanes which are able to split the data into classes. As a regression tool, the SVM offers flexibility to define the desirable margin of tolerance, wherein the error term is handled as a constraint and where the absolute error is to be equal or less than the maximum margin of tolerance \( \epsilon \). Often, the regression problem with SVM is stated as follows:

\[
\begin{align*}
\min_{w,b} & \quad \frac{1}{2} \|w\|^2 \\
\text{s.t.} & \quad |y_i - w \cdot x_i - b| \leq \epsilon \\
& \quad |w \cdot x_i + b - y_i| \leq \epsilon
\end{align*}
\]  \hspace{1cm} (33)

where, \( x_i \) is an input vector, \( y_i \) is its corresponding output vector, \( w \) is the weight vector, \( b \) is the bias, and \( w \cdot x_i - b \) denotes the prediction corresponding to sample \( x_i \). In practice, it is possible to use slack as penalties [38], as well as the Kernel trick to allow computing predictions with polynomial, Radial Basis Function (RBF) and sigmoid kernel functions.

Due to the fact of estimating the deflection error in three Cartesian coordinates, we aim here to find three regressors, each of which estimates the deflection error in an independent axis, as follows:

\[
\begin{align*}
f_{e_x} : q_a \rightarrow e_x, \\
f_{e_y} : q_a \rightarrow e_y, \\
f_{e_z} : q_a \rightarrow e_z,
\end{align*}
\]  \hspace{1cm} (34), (35), (36)

where, \( f_{e_x} \) denotes the SVM regressor model mapping the deflection of the passive joint angles \( q_a \) to the deflection error \( e_x \) in the x-axis.

3) Gaussian Process (GP)

GP is a nonparametric kernel-based probabilistic model that has been widely used for regression and classification prob-
points with similar predictor values. In supervised learning, generalization of unobserved values becomes possible under the mapping from input vectors to target values so that set based on prior knowledge of observed ones by learning [41]. As a regression tool, GP predicts the unobserved data full distribution to reflect the uncertainty in the prediction single prediction output for a training sample, GP offers the lems [40]. Unlike most machine learning models giving a single prediction output for a training sample, GP offers the full distribution to reflect the uncertainty in the prediction [41]. As a regression tool, GP predicts the unobserved data set based on prior knowledge of observed ones by learning the mapping from input vectors to target values so that generalization of unobserved values becomes possible under a provided uncertainty estimation. In supervised learning, points with similar predictor values $x_i$ are expected to have very close and similar target values $y_i$, which can be defined as covariance or a kernel function [42].

III. SIMULATION EXPERIMENTS

In order to evaluate the feasibility of estimating the position of the end-effector under flexibility conditions, we used simulation experiments to validate the effectiveness of the aforementioned models. The simulation was conducted on a dell desktop pc core i7 using SolidWork software 2021, MATA LB software 2021a and MSC ADAMS 2020. In this section, our computational experiments and results are described.

A. CONFIGURATION

The flexible version of the interconnected manipulator described in Fig 1 has been modeled and implemented in MSC ADAMS software which has been used as a virtual prototype, as shown in Fig 4. ADAMS (Automatic Dynamics Analysis of Mechanical Systems) as its name indicates has the ability to creates the dynamic motion equations automatically. so, in order to model the flexible manipulator, we proceed in a two-step approach. First, the manipulator was imported from SolidWorks to MSC ADAMS software as completely rigid mechanism. Then, the rigid links were replaced by flexible ones using the create FE part tool which can be found in the flexible bodies section of MSC ADAMS which is able to discretize these links for finite element analysis. Our goals in this regard are as follows:

- to introduce the flexible configuration parameters to the links,
- to collect the data that will be used in building and executing the machine learning algorithms.

Here, MSC ADAMS software serves as a virtual prototyping wherein the manipulator links are treated as flexible structures. Indeed, the dynamic analysis is carried out here using ADAMS software. MSC ADAMS is used to create trajectories for learning and testing machine learning schemes in addition to the actual end-effector position. It is also used as a simulator to get the robot response for the control schemes. ADAMS consider all dynamic parameters such as masses, moment of inertia, location of the mass-centers, flexibility of the links, input torques or motions, ...etc. For data collection, arbitrary values of the input angles within the workspace of $\theta_1$, $\theta_2$, and $\phi$ were applied to the active joints of the manipulator in order to obtain the angular deflection of the passive joints. Fig. 5 shows the kind of trajectories obtained, in which for each set of input values of $(\theta_1, \theta_2, \phi)$ of active joint angles, the corresponding values of the angular deflection of the passive joint angles were stored as tuples $q_a = (q_{a1}, q_{a2}, q_{a3}, q_{a4})$, as well as the localization error of the end-effector $e = (e_x, e_y, e_z)$. Fig. 6 shows a glimpse of the collected dataset for each independent measurement instance. The training-validation-testing ratio was defined as 60%-20%-20%. In practice, the angular deflections can be measured through optical encoders, yet in this paper, we measure them directly from the virtual prototype in MSC ADAMS software.

B. PREDICTION PERFORMANCE

This section presents the best-achieved results using the proposed machine learning methods to predict the actual end-effector position error of the flexible interconnected manipulator in the Cartesian space.

1) Artificial Neural Network (ANN)

In order to determine the suitable architecture and the relevant input mode for deflection sensing derived from the passive revolute joints, all possible NN configurations were evaluated. The mean square error $\text{MSE}$ and the correlation coefficient $R$ were used as performance metrics. The deployed NN was a feedforward architecture with eight layers,
six of which were hidden layers as shown by Fig. 7. The learning scheme consisted of the scaled conjugate gradient (SCG) due to its merits of space efficiency and scalability compared to other approaches such as the Levenberg-Marquardt (LM) [43]. Furthermore, we used the hyperbolic tangent sigmoid as a transfer function for hidden layers and a linear transfer function for the output layer. Several experiments and simulations have been conducted to determine the best number of hidden layers and the number of neurons in each hidden layer.

In order to show the performance of the NN, Fig. 8 shows the loss function ($MSE$) and the correlation coefficient ($R$) for all the 15 evaluated NN configurations. By observing Fig. 8, it is clear that all NN configurations can be used as a predictor for the error of the actual end-effector position. Particularly, the models with more than two inputs offer competitive prediction performance. For practical purposes, it is preferable to use model $f_{15}$ which recommends the setup of optical encoders at each passive joint. Thus, for the purpose of modeling the architecture and evaluating the performance frontiers further, model $f_{15}$ is considered for all experiments and simulations throughout this study.

To further validate the degree of effectiveness and accuracy of the NN models, different scenarios have been investigated and compared. First, the effect of changing the number of neurons was studied, then the effect of changing the number of the hidden layers. Also, the effect of using different NN architectures when the number of nodes in each layer show the ascending, descending, constant, and a bell-shaped configuration was studied. One of the main motivations of investigating the above scenarios is to obtain the suitable architecture to be deployed on-site, for which not only the number of nodes but also the network architecture become relevant [44].

### TABLE 1. The Performance in Terms of MSE, R and Training Time for Different Values of Neurons of Model $f_{15}$.

| Neurons | $MSE$-train $[10^{-3}]$ | $MSE$-test $[10^{-3}]$ | $R$ | Time (min.) |
|--------|----------------|----------------|----------|------------|
| 150    | 0.016          | 0.018           | 0.983    | 5.26       |
| 200    | 0.022          | 0.029           | 0.973    | 6.22       |
| 300    | 0.007          | 0.012           | 0.990    | 7.17       |
| 400    | 0.009          | 0.014           | 0.987    | 7.18       |
| 420    | 0.004          | 0.007           | 0.994    | 10.3       |
| 525    | 0.009          | 0.016           | 0.988    | 14.0       |
| 535    | **0.002**      | **0.004**       | **0.997**| **11.13**  |
| 600    | 0.004          | 0.009           | 0.993    | 15.51      |
| 760    | 0.005          | 0.007           | 0.993    | 21.08      |
| 825    | 0.005          | 0.005           | 0.993    | 22.0       |

Being the architecture using the four input modalities of the passive joints, Table 1 shows the performance of the model $f_{15}$ when changing the number of nodes in its architecture. The reader may note that Table 1 shows the number of nodes, the $MSE$ in both training-testing and the training time. Following the pyramidal distribution of the architecture shown in Fig. 7, we distributed the number of
nodes in a descending fashion. Based on the results and comparisons shown in Table 1, it was observed that all the NN models with more than 300 neurons achieved nearly the same regression performance. Naturally, scaling the network with a large set of nodes has an effect on increasing the training time in the order of minutes. Although the NN models with 300 and 420 neurons offer competitive MSE and correlation coefficient $R$, the model with 535 neurons offers the highest regression performance in training-testing. We also investigated the feasibility of using a different number of layers. Table 2 shows the performance when changing the number of layers of the best network model mentioned above. By observing Table 2, one can note that it is possible to achieve acceptable values of MSE and $R$ when using four hidden layers; however it is preferable to use six hidden layers due to the utmost performance. Furthermore, we investigated different approaches to distribute the 535 nodes in the NN architecture. In this scenario, we investigated four distinct architectures to distribute the neurons throughout the hidden layers, as shown in Table 3. By observing Table 3, one can note that distributing 535 neurons throughout the hidden layers in ascending orders offers the best performance in training-testing compared to the other distributions.

Based on the results, Fig. 9 shows the overall convergence performance of the NN model $f_{15}$ with 535 nodes under an ascending order of node configuration across six layers. By observing Fig. 9 and Table 3, one can observe the performance of the aforementioned NN achieves $0.008 \times 10^{-3}$ at epoch 6591 during the validation period, and $0.005 \times 10^{-3}$ at the end of training phase, and $0.006 \times 10^{-3}$ on the testing phase. These results represent the best overall performance of the above studied NN architectures. In addition, Fig. 9-(b) illustrates the regression and correlation analysis for trained, validated, and testing datasets. In this figure, if the ANN output is similar to the target, then the line fitting the output and target will have a $45^\circ$ inclination and a correlation coefficient $R = 1$. By observing this figure, one can note that the high ratios of correlation coefficient, all of which
are proximal to 1. These results show the appealing features of the NN models to estimate the deflection error based on the measurements of the passive joint angles. Although it is possible to use a larger number of nodes to achieve improved performance, the study of larger and deeper networks is out of the scope of this paper.

2) Support Vector Machine (SVM)

We also investigated the feasibility of using SVM as a regression mechanism. Here, we implemented the Support Vector Regression (SVR) approach, wherein three independent SVR models were used to predict each element of the deflection \( \epsilon \) in each axis. We used the training-testing ratio of 80%-20%, respectively, and evaluated the linear, the polynomial (quadratic and cubic), and RBF, as kernels.

For the sake of using optimal parameters for each kernel function, we used the Bayesian Optimization as a hyper-parameter tuning in MATLAB, achieving the best model parameters with the lowest MSE performance. The motivation of using Bayesian Optimization is due to the surrogate-based approach to search optimal parameters, which implies the computationally expensive objective function evaluation. Evaluating and comparing the performance of other optimization algorithms for hyperparameter tuning is out of the scope of this paper.

In order to evaluate the performance of the models obtained by SVR, Fig. 10 shows the prediction performance when using distinct types of kernel functions. By looking at the results of Fig. 10, we can observe that the RBF offers the utmost competitive performance when predicting the deflection error in the three axes. The parameters of the RBF-based model tuned by Bayesian Optimization are shown in Table. 4. The first parameter is the box constraint \( C \), which is a penalty factor on miss-regressions, the obtained large values of \( C \) imply the stricter regression. The second parameter is the \( \epsilon \) factor, in which the relatively small values imply the small margin of tolerance for regression. The third parameter is the kernel scale \( \gamma \); the relatively low obtained values imply the tendency to downscale the input data.

3) Gaussian Process (GP)

We also investigated the feasibility of GP as regression models. The training-testing ratio was 80%-20%, respectively. In line with the motivations behind hyperparameter tuning in SVM, we used Bayesian Optimization to search for optimal parameters in GP models. Since the kernel function in GP determines the similarity between two inputs, we studied the regression performance of relevant kernel functions such as the Matern 5/2, the rational quadratic function, the exponential, and the squared exponential functions.

To evaluate the performance of the models learned by GP, Fig. 11 shows the prediction performance for each kernel function used. By observing this figure, one can note that the GP with exponential kernel function offers the best perfor-
TABLE 5. The Optimized Parameters for the Three GP Models.

| Parameter                        | I_e_f | I_e_g | I_e_h |
|----------------------------------|-------|-------|-------|
| Signal standard deviation (\(\sigma_f\)) | 0.3187 | 0.4736 | 0.5378 |
| Characteristic length scale (\(\sigma_l\)) | 0.0460 | 0.0460 | 0.0460 |

TABLE 6. Comparison Between the Three Proposed Models: (ANN, SVM, GP) According to MSE, and R.

| Model                | MSE   | R    |
|----------------------|-------|------|
| NN                   | 0.004 | 0.996|
| SVM (Gaussian/RBF)   | 0.01297 | 0.964|
| GP (Exponential)     | 0.006 | 0.98 |

mance to predict the deflection error in the three Cartesian coordinates. The learned parameters of the GP with exponential kernel function are shown by Table 5 in which the small values of the signal standard deviation \(\sigma_f\) denotes a relatively small range magnitude able to represent by the kernel function, and the small values of the characteristic length scale \(\sigma_l\) shows the tendency to scale down the distances between inputs.

C. MODEL TESTING

To evaluate the generalization ability of the obtained models, in this section, we describe the performance in unseen scenarios, that is, trajectories that were not used during the training phase.

1) Testing Phase

It is possible to compare the overall performance of the evaluated models in the testing scenarios, Table 6 shows the performance of the best models obtained from the testing point of view. As such, by observing Table 6, one can note that all of the models offer reasonable performance in terms of MSE metric and R coefficients during the testing scenarios, implying that NN, SVM, and GP are effective for deployment purposes. However, due to a large number of nodes/parameters, we observed that the models derived from NN were computationally more efficient during the training stage compared to SVM and GP.

2) Out-of-sample testing

The dataset used for training-testing consists of arbitrary trajectories within the workspace of the manipulator as shown by Fig. 5, whose corresponding arc lengths are relatively short compared to the entire workspace volume. In this line, the testing dataset, which is an arbitrary subset of the entire dataset poses the risk of providing an erroneous mechanism to evaluate the generalization performance and overfitting challenges of the studied function approximation schemes.

In order to rigorously evaluate the generalization ability of the obtained function approximation schemes, we used manipulator trajectories that are structurally and geometrically different than the ones derived from the training-testing approach. In this context, we used manipulator trajectories being relevant to real-world manipulation and handling applications, some of which are the following:

- a trajectory with segments, in particular one with two vertical segments and one horizontal segment to simulate the pick and place tasks in industrial domains, such as the one shown in the top side of Fig. 12.
- a spiral trajectory to simulate the motion in repetitive tasks with constant curvature considerations yet arbitrary orientation in the 3D domain, such as the one shown in the top side of Fig. 13.
- a lemniscate-based trajectory to simulate the varying nature of the curvature in the 3D domain, such as the one shown in the top side of Fig. 14.
- a curvilinear trajectory in 3D domain to simulate the motion between two arbitrary points in the workspace while avoiding obstacles, such as the one shown in the top side of Fig. 15.
- an ellipsoid-based trajectory on simulating the motion where the origin and destination are the same point in the 3D domain, such as the one shown by Fig. 16.

Accordingly, the procedure to compare the performance by using out-of-sample trajectories were as follows:

- Once the training phase is realized, the angular deflection of the passive joint angles \(q_a = (q_{a1}, q_{a2}, q_{a3}, q_{a4})\) corresponding to the above-mentioned trajectories are used as inputs for the function approximation scheme to estimate the deflection error \(e\), as Fig. 3 shows.
- Then, to compute the end-effector position, we used the deflection error \(e\) along with the estimation of the FK, as Fig. 3 shows.
- At the same time, the manipulator is given input signals to the active joint angles \(q = (\theta_1, \theta_2, \phi)\) to generate the above-mentioned trajectories.
- Afterwards, a comparison between the generated end-effector position and the actual one obtained from MSC ADAMS software is performed for NNs, SVM, and GP. For simplicity and without loss of generality, we used the best models from the previous section.

In order to show the overall comparison of the prediction performance of the obtained models in each out-of-sample trajectory, Fig. 12 - Fig. 16 show the performance of the best-obtained model of NN, SVM, and GP in all out-of-sample trajectories. These figures show the comparison and degree of overlapping between the actual and the generated trajectories as well as the error between the actual and the
FIGURE 12. Performance in out-of-sample scenario inspired on pick and place. Top: error between the actual and the generated trajectory, bottom: error in the three Cartesian coordinates as a function of simulation time. The performance of: (a) NN model, (b) SVM, and (c) GP.

FIGURE 13. Performance in out-of-sample scenario based on a spiral trajectory. Top: error between the actual and the generated trajectory, bottom: error in the three Cartesian coordinates as a function of simulation time. The performance of: (a) NN model, (b) SVM, and (c) GP.

generated trajectory in the Cartesian coordinates with respect to the simulation time. By observing the results in these figures, one can note that all models offer the reasonable and effective estimation of the end-effector position under distinct unseen scenarios, implying the feasibility for use in practical manipulation tasks. One can also note that, the NN model offers a smaller magnitude of error with respect to simulation time along with the three Cartesian coordinates across all scenarios, implying the more accurate prediction of end-effector position under flexibility considerations. In fact, we noted that the maximum error of the NN model in all out-of-sample scenarios did not exceed $0.15 \times 10^{-3}$ m. These results are due to the larger number of parameters/nodes of the NNs, which are in line with the results shown in Table 6, portraying the promising results of the NN model in the testing phase.
In order to characterize and summarize the performance of the learned models in each out-of-sample trajectory, Fig. 17 shows the performance comparison in terms of MSE metric of the best-obtained model of NN, SVM, and GP in each aforementioned out-of-sample trajectory. By looking at Fig. 17, we can observe that the NN model is able to achieve lower MSE values during the prediction task. These results are in line with the above-mentioned performances, showing that owing to the improved prediction of deflection error leads to lower error in the prediction of the position of the end-effector under flexibility considerations.

The above-mentioned results show that either of the obtained NN, SVM, and GP models can be used to achieve reasonable performance in the prediction of the position of the end-effector. However, the NN will be preferable in situations implying the finer and more accurate estimation. Although
it is possible to devise a larger number of out-of-sample testing trajectory scenarios, their study is out of the scope of this paper, and we rather focused on the above-mentioned set due to their relevance for straightforward application to manipulation tasks in industrial scenarios. This work aims to estimate the position of the end-effector. So, only the trajectories of the end-effector are shown as in Figures (12-16). The length of the trajectories are hundreds of centimeters while the small deflection are few millimeters and maybe superimposed on the large motion along the trajectory in a smooth way. Because of that, they are not visible in the 3D graphs in the same figures. However, these small deflections are estimated and shown at different instances as in Figure 6(h-j).

D. APPLICATION IN CONTROL DESIGN

The function approximation scheme is useful to enable the closed-loop control mechanisms that consider the deflection error due to flexibility considerations. Thus, this section presents the application of the function approximation scheme in the design of a control mechanism for the flexible manipulator. In this line, observations in the joint space and the task space control are presented.

1) Joint Space Control

Here, the manipulator is expected to reach (or track) a desired target in the joint space. However, since the conventional control in the joint space is unable to consider the deflection errors due to flexibility, the conventional approach will be unable to reach (or track) the desired target. To give an example of this assertion, Fig. 18 shows the basic idea of a control scheme based on the joint space. For simplicity, we use the proportional–integral–derivative (PID) control approach since the PID controller is still considered the most widely used control algorithm in industrial control applications [45]; the study of advanced control schemes is out of the scope of this paper. In addition, By observing Fig. 18, the desired target \( p_d = (X_d, Y_d, Z_d) \) is first mapped into the joint space using the inverse kinematics, then the PID control is designed in the torque space based on the information obtained from the joint space. The output of PID control is used to determine the suitable torques \( \tau = (\tau_1, \tau_2, \tau_3) \) to actuate the active joints \( q = (\theta_1, \theta_2, \phi) \) to reach the target values \( p_d = (X_d, Y_d, Z_d) \) (previously computed by the inverse kinematics). The actual position \( p = (X, Y, Z) \) of the joint angles is consequently obtained from the FK. Naturally, since the FK assumes rigid links, the desired target \( p_d = (X_d, Y_d, Z_d) \) is expected to be different from the actual end-effector position \( p = (X, Y, Z) \) due to the flexibility considerations of the links of the manipulator.

In line with the above-described points, in order to examine the accuracy of the joint space control in a flexible-
Based on the interconnected manipulator, the coordinate $p^d = (800, 0, -100)$ mm was chosen as a target point in the Cartesian domain. Fig. 19 shows the performance of the joint space control in the three axes, as well as the magnification over the convergence period. By observing the results in this figure, one can note that the joint space control is able to reach the target goal of each axis. However, when one compares the actual position of the end-effector and the achieved position by the controller, it is straightforward to note the error/discrepancy in the right side of the figure. This observation is mainly due to the unconsidered flexibility of the manipulator structure. As shown in Fig. (18), the conventional joint space control uses conventional inverse and forward kinematics which treats the manipulator as a rigid one. This type of control techniques uses encoders located at the active joints to calculate the end-effector position assuming rigid links. The actual position of the end-effector that is affected by the flexibility of the links cannot be sensed or estimated. So, the conventional joint space control is unable to reach the desired target with utmost accuracy. Although we used a target point to exemplify the above-described notion, it is possible to use trajectories to further evaluate the joint space control. Such an approach, however, is out of the scope of this paper since the deflection errors due to the flexibility of the link are not considered explicitly.
2) Task Space Control

On the other hand, the task space control is designed differently from the joint space control since it is developed directly in the operational space. Therefore, it is more sensitive to the flexibility error than the joint space [46]. For simplicity, we used the inverse Jacobian-based control to show the applicability of the function approximation schemes when designing flexibility-oriented control schemes.

In line with the above consideration, the conventional inverse Jacobian-based control, as shown in Fig. 20, has been extended to consider the function approximation to estimate the deflection error occurring due to flexible links. In Fig. 20-(a), for a desired target \( \mathbf{p}^d \), it is possible to estimate \( \delta \mathbf{p} \) by using the target \( \mathbf{p}^d \) and the actual position \( \mathbf{p}^r \), then it is possible to estimate \( \delta \mathbf{q} \) by using the inverse Jacobian \( \mathbf{J}^{-1} \). Then, the control input generalized torques \( \tau \) can be computed based on \( \delta \mathbf{q} \) and a suitable feedback matrix gain. As such, the actual position \( \mathbf{p}^r \) can be estimated by the FK. However, the deflection error due to flexible links in the manipulator cannot be measured directly through the FK. Therefore, the function approximation scheme has been implemented to estimate the actual position \( \mathbf{p} = (X, Y, Z) \) of the manipulator as shown in Fig. 20-(b).

As for the implementation of the above approach:

- The CAD model of our flexible interconnected manipulator was imported in MSC ADAMS multibody dynamics software to introduce the flexible parameters to the links of the manipulator.
- Then, a co-simulation between MSC ADAMS and MATLAB/Simulink has been carried. As outlined in the previous sections, the prototype in MSC ADAMS serves as a virtual environment that considers flexibility considerations.
- The control algorithm consisted of a Proportional-Integral-Derivative (PID) approach.

To show the feasibility of using the function approximation scheme to design control approaches that consider flexible manipulators, we used the NN model with 535 nodes, 6 layers and an ascending distribution of nodes due to the improved generalization ability discussed in section III-C. Using other models based on SVM or GP is straightforward. Also, the evaluation in the relevant hardware and experimental setup is left for future work. For desired target trajectory, a sine function with an amplitude of 200 mm and frequency of 0.5 rad was designed to move the end-effector in the X-axis and Y axis while, a sine function with an amplitude of 100 mm and frequency of 0.5 rad was designed to move the end-effector in the third axis (Z). The performance of the control in the task space was tested in a simulation with total time of 10 seconds and a payload of 0.5 kg applied at the end-effector. The initial active joint angles were fixed at \( \theta_1 \), \( \theta_2 \), and \( \phi \) as \( 48^\circ \), \( 130^\circ \) and \( 0^\circ \) respectively, to avoid the singular configuration at the start of the simulation.

To exhibit the performance of the control approach using the function approximation scheme to tackle the flexibility considerations, Fig. 21 shows the comparison between the desired target trajectory and the (achieved) actual trajectory of the end-effector of the flexible manipulator, while Fig. 22 shows the errors in the three axes. By observing the results of Fig. 21 and Fig 22, one can note that the proposed control approach is able to follow the desired trajectory with a very small error that does not exceed \( 1 \times 10^{-3} \) m in the three axes(X, Y, Z). These results are in line with our findings in section III-C, implying that the explicit consideration and improved estimation of the deflection error due to flexibility leads to better estimation and tracking of the position of the
IV. CONCLUSION

This paper has investigated the application of the virtual sensor theory in combination with function approximation schemes derived from the machine learning literature to estimate the position of the end-effector of a flexible industrial interconnected manipulator. MSC ADAMS software has been used as a virtual prototype where the manipulator links are treated as flexible links. Since the manipulator has a complex structure, it is unfeasible to create the mathematical equations that are able to describe the final position of the flexible interconnected robot in the Cartesian space.

To tackle the above-mentioned problem, we rigorously evaluated the feasibility of using machine learning frameworks such as NN, SVM, and GP to estimate the deflection error based on the values of the angular deflection of the passive joint angles as inputs. Also, we rigorously evaluated the generalization abilities of NN, SVM, and GP in predicting the trajectory of the manipulator in diverse environments being relevant to real-world applications, that is, trajectories comprising a pick and place task, a spiral, a lemniscate, a curved trajectory, and an ellipsoid. Our rigorous computational experiments have shown that: (1) the NN, SVM, and GP models are able to attain promising prediction accuracy in estimating the position of the interconnected manipulator, and (2) a feedforward NN with 535 neurons with six hidden layers in which nodes are distributed in an ascending approach offers the improved prediction and generalization to unseen environments; implying the enhanced and robust prediction of the position of the end-effector under flexibility considerations (the upper bound on the achieved error did not exceed $0.15 \times 10^{-3}$ m). Furthermore, we evaluated control schemes under flexibility considerations: (1) we showed that the conventional control in the joint space was unable to reach (or track) the desired target due to being unable to consider the deflection errors which occur due to the flexible manipulator, and (2) we studied the feasibility of using a function approximation scheme to design a control approach that explicitly considers the deflection error flexibility considerations. As such, we outlined a control approach based on the inverse Jacobian, and used NN-based function approximation to estimate the deflection errors. Our computational experiments show that it becomes possible to follow a sinusoidal trajectory with reasonable performance.

Our results have shown the effectiveness of the nonlinear
relationships learned by NN, SVM, and GP to predict the position of the flexible manipulator with promising/desirable accuracy. In future works, we plan to study the possibility of performing online training in dynamic environments besides the dynamical analysis of the manipulator in order to analyze the nonlinearities and interactions of industrial robots, as well as study the performance of advanced control architectures to solve some of the problems associated with the flexibility.

REFERENCES

[1] A. Singla and A. Singh, Dynamic modeling of flexible robotic manipulators, vol. 741, Springer Singapore, 2019.
[2] Tsai, Lung-Wen, Robot analysis: the mechanics of serial and parallel manipulators, John Wiley & Sons, 1999.
[3] Klimchuk, Alexander, Enhanced stiffness modeling of serial and parallel manipulators for robotic-based processing of high performance materials, École Centrale de Nantes (ECN); Ecole des Mines de Nantes, 2011.
[4] 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 (Switzerland), vol. 17, no. 9, pp. 1–22, 2017, doi: 10.3390/s17091934.
[5] Angeles, Jorge, and J. Angeles, Fundamentals of robotic mechanical systems. Vol. 2, New York: Springer-Verlag, 2002.
[6] P. Pagis, N. Bouton, S. Briot, and P. Martinet, “Design of a controller for enabling parallel robots workspace through Type 2 singularity crossing,” 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014.
[7] Theodore, Rex J., and Ashitava Ghosal, “Comparison of the assumed modes and finite element models for flexible multilink manipulators,” The International journal of robotics research 14.2 (1995): 91-111.
[8] A. M. Shafei, M. H. Korayem, “Theoretical and experimental study of DLCC for flexible robotic arms in point-to-point motion,” Optimal Control, Applications and Methods, Vol. 38, No. 6, pp. 963-972 (2017).
[9] Ebrahimi, S. Eshaghieh-Firoozobadi, “Dynamic performance evaluation of serial and parallel RPR manipulators with flexible intermediate links,” Iran. J. Sci. Technol. Trans. Mech. Eng. 2016, 40, 169–180.
[10] M. Sharifinia, A. Akbarzadeh, “A constrained assumed modes method for dynamics of a flexible planar serial robot with prismatic joints,” Multibody Syst. Dyn. 2016, 44, 49–65.
[11] Usoro, P.B. and Nadira, R. and Mahil, S.S. “A finite element / Lagrange approach to modeling lightweight flexible manipulators,” ASME J. Dynamic systems Meas. Control., 108, pp 198-205 (1986).
[12] Geniele, Howard, Rajnikant V. Patel, and Khashayar Khorasani, “Control of a flexible-link manipulator.” Proceedings of 1995 IEEE International Conference on Robotics and Automation. Vol. 1. IEEE, 1995.
[13] M. H. Korayem; A. M. Shafei; F. Absalan; B. Kadkhodaei; A. Azimi, “Kinematic and dynamic modeling of viscoelastic robotic manipulators using Timoshenko beam theory,” theory and experiment, International Journal of Advanced Manufacturing Technology, 71, 5-8, 1005-1018, 2014.
[14] M Ahmadizadeh, AM Shafei, R Jafari, “Frictional Impact-Contacts in Multiple Flexible Links,” International Journal of Structural Stability and Dynamics, 21 (06). 2150075.
[15] M. H. Korayem, A. M. Shafei, S. F. Dehkhordi, “Systematic Modeling of a Chain of N-Flexible Link Manipulators Connected by Revolute-Prismatic Joints Using Recursive Gibbs-Appell Formulation”, Archive of Applied Mechanics, Vol. 84, No. 2, pp. 187-206, 2014.
[16] Nagaraj B P, Nataraju B S, A. Ghosal, “Dynamic of a two-link flexible manipulator,” IEEE Transactions on Robotics and Automation Society, pp. 207-220, 1997.
[17] Kiang, Chang Tai, Andrew Spowage, and Chan Kuan Yoon, “Review of Control and Sensor System of Flexible Manipulator,” Journal of Intelligent & Robotic Systems. Theory Appl., vol. 77, no. 1, pp. 187–213, 2015.
[18] L. Bascetta and P. Rocco, “End-point vibration sensing of planar flexible manipulators through visual servoing,” Mechatronics, vol. 16, no. 3–4, pp. 221–232, 2006.
[19] Dubus, Gregory, “On-line estimation of time varying capture delay for vision-based vibration control of flexible manipulators deployed in hostile environments,” 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2010.
[20] M. S. Saad, H. Jamaluddin, and J. Z. M. Darus, “Active vibration control of a flexible beam using system identification and controller tuning by evolutionary algorithms,” JVC/Journal Vib. Control, vol. 21, no. 10, pp. 2027–2042, 2015, doi: 10.1177/1077546313505635.
[21] G. Dubus, O. David, and Y. Measson, “A vision-based method for estimating vibrations of a flexible arm using on-line sinusoidal regression,” IEEE Robotics and Automation Society, pp. 4068–4075, 2010.
[22] Liu, Hongyan, et al, “Design of adaptive fuzzy controller for flexible link manipulator,” IEEE International Conference on Industrial Technology. IEEE, 2008.
[23] M. Magdy, M. Fanni, A. M. Mohamed, and T. Miyashita, “Kinematic design and novel mobility analysis of a new 3D pantograph decoupled manipulator,” Mechanism and Machine Theory 117 (2017): 253-275.
[24]Sameh, Ahmed, Mohamed Fanni, and Abdelfatah M. Mohamed, “New 3D Translational Interconnected Manipulator for Industrial Applications,” 2018 IEEE International Conference on Mechatronics and Automation (ICMA), pp. 469-474. IEEE, 2018.
[25] Hill, Rafael Balderas, Design of high-speed robots with drastically reduced energy consumption, Diss. Ecole centrale de Nantes, 2019.
[26] Rognant, Mathieu, Eric Courtelle, and Patrick Maurice, “A systematic procedure for the elastodynamic modeling and identification of robot manipulators,” IEEE Transactions on robotics 26.6 (2010): 1085-1093.
[27] M. H. Ettefaghi; M. Naraghi; M. Mahzoon, “Experimental control of a flexible link by the method of Controlled Lagrangian,” Journal of theoretical and applied vibration and acoustics, Vol. 4, No. 1, 81-98, 2018.
[28] E. Mostafa, A. M. Mohamed, “concurrent design for gravity balancing and rigidity enhancing of a new interconnected manipulator,” IMECE2021-73333, pp. 1–10, 2021.
[29] Sameh, M. Fanni, V. Parque, and M. Mohamed, “Development of a Balanced 3D Translational Interconnected Manipulator With Solely Rotary Joints/Actuators and Free-Internal-Singularity Workspace,” IEEE Access 9 (2021): 167880-167899.
[30] M. Magdy, M. Fanni, M. Mohamed, and T. Miyashita, “Kinematic design and novel mobility analysis of a new 3D pantograph decoupled manipulator,” Mechanism and Machine Theory, vol. 117, pp. 253–275, 2017.
[31] A. Sameh, M. Fanni, and A. M. Mohamed, “Finite Element Analysis, Control and Simulation of a Novel 3D Hybrid Balanced Manipulator,” no. 01, pp 9–19, 2021.
[32] S. Briot and J. A. Bonev, “Pantopteron-4: A new 3TR decoupled parallel manipulator for pick-and-place applications,” Mechanism and Machine Theory, vol. 45, no. 5, pp. 707–721, 2010.
[33] B. Siciliano, “Kinematic control of redundant robot manipulators: A tutorial,” Journal of intelligent and robotic systems., vol. 3, no. 3, pp. 201–212, 1990.
[34] A. Jamali, I. Z. M. Darus, P. P. S. Samin, and M. O. Tokhi, “Intelligent modeling of double link flexible robotic manipulator using artificial neural networks,” Journal of Engineering Design, pp. 1–15, 2018.
[35] Bingul, Z., H. M. Ertunc, and C. Oyus, “Applying neural network to inverse kinematic problem for 6R robot manipulator with offset wrist,” Adaptive and Natural Computing Algorithms. Springer, Vienna, 2005. 112-115.
[36] Z. R. Yang, “Artificial Neural Networks,” Comprehensive Biomedical Physics, Elsevier Pages 1-17,2014.
[37] Xu, Xiang, and Daoli, “New method for solving Ivanov regularization-based support vector machine learning,” Computers & Operations Research 136 (2021).
[38] V. N. Vapnik, “The Nature of Statistical Learning,” Theory. p. 334, 1995, [Online]. Available: http://arxiv.org/abs/2009.10862.
[39] V. Vapnik, S. E. Golowich, and A. Smola, “Support vector method for regularization-based support vector machine learning,” Computers & Operations Research, Pages 1-17, 2014.
[40] J. Wang, “An Intuitive Tutorial to Gaussian Processes Regression,” 2020, [Online]. Available: http://arxiv.org/abs/2009.10862.
[41] P. Gardner, T. J. Rogers, C. Lord, and R. J. Barborthe, “Learning model discrepancy: A Gaussian process and sampling-based approach,” Mech. Syst. Signal Process., vol. 152, p. 107381, 2021.
[42] Rasmussen, Carl Edward, and C. Williams, “Gaussian processes for machine learning the mit press.” Cambridge, MA 32 (2006).
[43] S. Mishra, R. Prusty, and P. K. Hota, “Analysis of Levenberg-Marquardt and Scaled Conjugate gradient training algorithms for artificial neural network based LS and MMSE estimated channel equalizers,” Proc. - 2015 Int. Conf. Man Mach. Interfacing, MAMI 2015, no. December 2015, 2016.
[44] Y. A. Al-sbou and K. M. Alawasa, “Nonlinear Autoregressive Recurrent Neural Network Model For Solar Radiation Prediction,” vol. 12, no. 14, pp. 4518–4527, 2017.
[45] Yu, Cheng-Ching, “Autotuning of PID controllers: a relay feedback approach,” Springer Science & Business Media, 2006.
[46] Bruno, Siciliano, Sciavicco Lorenzo, Villani Luigi, and Oriolo Giuseppe, “Robotics: modelling, planning and control,” (2010).

MUHAMMAD ADEL was born in Zagazig, Egypt, in 1994. He received the B.Sc. from the Mechanical Engineering Department, Faculty of Engineering, Zagazig University, in 2012. In 2018, he received the M.Sc Scholarship from the Egypt-Japan University of Science and Technology. His areas of interests include machine learning, Industrial Robots, Design and Control of Mechatronic Systems.

SABAH M. AHMED was born in Assiut, Egypt, in 1956. She received B.Sc. (with honors) and M.Sc. degrees in electrical engineering from Assiut University, Egypt, in 1979 and 1985 respectively. She received her Ph. D. degree from the Technical University of Budapest, Hungary in 1992. She has worked at the Faculty of Engineering, Assiut University as a Demonstrator (1979), an Assistant Lecturer (1985), an Assistant Professor (1992), an Associate Professor (2002) and a Professor (2009). She joined the Electronics Engineering Department, Hijjawi Faculty for Engineering Technology, Yarmouk University (1996), Jordan; Jarash Private University (1997), Jordan; and Computer Science Department, Zarka Private University (2000), Jordan, as a Visiting Professor. Professor Sabah research interests include Speech analysis and synthesis, digital filters, biomedical signal processing, data compression, wavelet transforms, genetic and immune algorithms, and mechatronics and robotics. She authored and co-authored over 60 published articles in the above fields. She is currently a Professor in Mechatronics and Robotics Engineering Department, School of Innovative Design Engineering, Egypt-Japan University of Science and Technology (E-Just), Egypt, Since August 2017.

MOHAMED FANNI received the B.E. and M.Sc. degrees in mechanical engineering from Faculty of Engineering of both Cairo University and Mansoura University, Egypt, in 1981 and 1986, respectively and the Ph.D. degree in engineering from Karlsruhe University, Germany, 1993. He is a Professor with the Dept. of Mechatronics and Robotics Engineering Egypt-Japan University of Science and Technology E-JUST, Alexandria, on leave from Production Engineering and Mechanical Design Department, Faculty of Engineering, Mansoura University, Egypt. His major research interests include robotics engineering, automatic control, and Mechanical Design. His current research focuses on Design and Control of Mechatronic Systems, Surgical Manipulators, Industrial Robots and Flying/Walking Robots.