IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 11, 2022, accepted April 18, 2022, date of publication April 29, 2022, date of current version May 6, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3171236

Joint-Space Kinematic Control of a Bionic Continuum Manipulator in Real-Time by Using Hybrid Approach

MRUNAL KANTI MISHRA, ARUN KUMAR SAMANTARAY, AND GOUTAM CHAKRABORTY
Department of Mechanical Engineering, Indian Institute of Technology Kharagpur, Kharagpur 721302, India

Corresponding author: Arun Kumar Samantaray (samantaray@mech.iitkgp.ac.in)

This work was supported in part by the Indo French Centre for the Promotion of Advanced Research (IFCPAR), under Department of Science and Technology (DST) India and Centre National de la Recherche Scientifique (CNRS) France Collaboration for the Project "Modelling and Control of Mobile Cooperating Bionic Arms" Vide, under Grant DST-CNRS 2015-02.

ABSTRACT Continuum manipulators are a type of robot used for delicate applications, including safe human-robot interactions. Controlling these manipulators for an accurate trajectory, especially in the case of pneumatic actuation, is extremely challenging. Thus, this article proposes a real-time kinematic trajectory control of a pneumatically actuated multi-segment bionic continuum manipulator with a mobile base by combining a neural network and analytical model with a cascaded controller to overcome this challenge. The inverse kinematics solution of the multi-segment manipulator is developed by using a neural network and an inverse piecewise constant curvature approach. The neural network is trained by using a separate learning algorithm. Although hybrid inverse modeling gives better solutions than existing techniques, significant residual positional error of the manipulator tip remains due to inherent material hysteresis. Thus, a cascaded PI-controller is utilized to compensate for the residual positional error. The controller gains are updated in each step by predictions of the actuator length, where the Jacobian entries are computed from the neural network model. The proposed procedure is validated on Festo Didactics' elephant trunk-like two-segment continuum manipulator, Robotino-XT. Three different cases are considered for real-time trajectory tracking, where the OptiTrack vision system is used for validation by tracking the manipulator tip pose. For the trajectory points outside the manipulator workspace, simultaneous trunk and base movements are used. In experimental validation, the proposed scheme is shown to give much reduced manipulator tip trajectory error as compared to the existing methods.

INDEX TERMS Bionic continuum manipulator, inverse kinematics, kinematic control, neural network, separate learning algorithm.

I. INTRODUCTION

Nowadays, there is a growing research interest in bionic continuum manipulators (BCMs) which are inspired by different living organisms and try to emulate their structures and motions, such as snake [1], octopus arm [2], and elephant trunk [3] (See Fig. 1) robots. BCMs are replacing conventional rigid robots in specific fields where versatility and safety of human-robot interaction are prioritized over speed and positional accuracy [4]. Since continuum manipulators are highly dexterous, they are used in cluttered environments and delicate applications such as invasive medical surgery [5], rescue operation [6], and fish handling. For such manipulators, various actuation principles like pneumatic actuated muscles [7]–[9] and tendon drives [10] are utilized to accomplish large movements.

Despite several advantages, these under-actuated robots suffer control inaccuracies [11]. The nonlinear material behavior such as hysteresis, especially under large strain, compromises repeatability and adds complexity to the interpretation and prediction of the system behavior [12]. This unpredictability is a significant challenge that has drawn the attention of a significant number of researchers.

The kinematic modeling procedures of a BCM are much more complex than their rigid counterparts due to the high flexibility of the links. Moreover, the conventional modeling

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

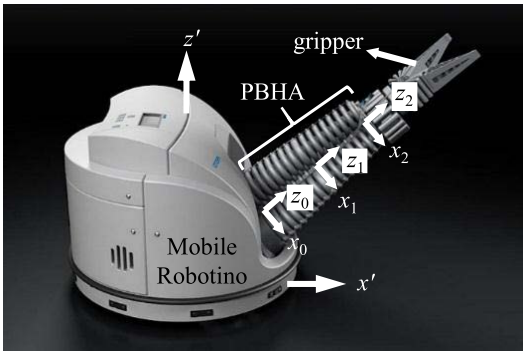

FIGURE 1. Robotino-XT (image courtesy: "Festo AG & Co. KG., all rights reserved"; reproduced with permission).

approaches cannot be applied to a BCM due to the lack of discrete rigid links in their structure [13]. The existing modeling approaches of the BCMs are broadly based on either the constant curvature model ([8], [14]–[18]), variable constant curvature model [20], or variable curvature model [21]–[26]. In the constant curvature model, the manipulator backbone is assumed to be a perfectly circular arc at any state [15]. The developed model in [15] is extended for a multi-segment manipulator to increase dexterity by considering C_0 and C_1 continuity between the segments [16], [17]. In recently reported applications of the constant curvature modeling approach, a multi-segment BCM is controlled for invasive medical surgery [16], [27]. Similar to the constant curvature model, the piece-wise constant curvature model is developed by considering a large number of small segments connected in series [20]. However, both constant and piece-wise constant curvature modeling approaches fail to accurately evaluate the end-effector pose of non-cylindrical cross-section manipulator segments, such as conic or bellow-type manipulator segments. Variable curvature models are developed to fit the backbone shape of a conic BCM for better tip accuracy [22], [25]. Similar to the geometric constant curvature model, the variable curvature model is developed for conic manipulators [22]. The elementary difference in both models is the consideration of exact actuator length. In another geometric modeling approach, the Pythagorean hodograph approach is used to map the length of a conic BCM to tip pose [23]. The authors in [25] proposed an exact analytical forward kinematic (FK) model for a single segment conic BCM based on an elliptic integral approach. On the other hand, the Cosserat-rod method, which deals explicitly with continuum structure, computes the most accurate results for the FK modeling of BCMs by employing numerical simulations [26]. However, most of the techniques mentioned above lack accuracy in task space due to several assumptions and are computationally complex for real-time implementation [28].

High precision model-based inverse kinematics (IK) for BCMs remains a challenge. While the differential Jacobian method [3], [29] is most often used, resolving the redundancy for highly under-actuated continuum manipulators poses several challenges. The modal approach [30] and the elastic beam theory approach [31] are regarded as the most accurate methods to map task space parameters (TSPs) to joint space parameters (JSPs) by considering linear combinations of the shape functions. However, those perform poorly during real-time control of the hyper-redundant robots due to high computational complexity and delayed feedback. In [32], a geometric IK approach of a pneumatically actuated BCM is proposed, where each segment's pose is assumed *a priori*. However, it is desired to find all the JSPs (manipulator actuator lengths) from only the target pose in most practical applications. Moreover, most of the above-mentioned model-based approaches fail to find the JSPs accurately, especially for the applications that need precise solutions. Moreover, most existing modeling techniques do not offer a solution to avoid singularity problems. It is also a tedious job to find the inverse solutions from the FK because of the limitations of the numerical methods, which tend to be too slow for real-time implementation [28].

Thus, in recent times, model-free approaches to calculate the inverse solutions of the BCMs have gained in popularity. The JSPs of a two-segment conic BCM is approximately estimated from the TSPs (manipulator pose) by using distal supervised learning and radial basis function in the neural network (NN) model [33]. The work in [33] is extended in [34] to formulate an adaptive NN model for improved trajectory tracking. In [35], the goal babbling approach is developed to tackle the IK problem of a three-segment continuum manipulator. A hybrid method for the evaluation of IK of compact bionic handling assistant is proposed in [36]. In [26], a feed-forward NN model is used to approximate the inverse solution of a BCM. Considerably high accuracy is reported for shape reconstruction of a BCM under external loading conditions by using short-term memory NN [37]. The above-discussed inverse models work in well-defined environments and lack robustness, i.e., accurate trajectory tracking capability under unknown external disturbances acting on the manipulator backbone. Thus, there is a need to develop adaptive kinematic control approaches for accurate trajectory tracking of BCM in the presence of uncertainties.

Different control strategies have been developed to control the tip pose and shape of BCM. A closed-loop task-space controller, explicitly specifying the shape configuration, is proposed for a tendon-driven BCM [38]. Therein, IK is solved as a nonlinear optimization problem. An arc space (circular arc parameters of each segment) controller is proposed in [39], which uses external data of each segment's configuration parameters and actuator lengths to achieve asymptotic tracking of arc parameters. In [40], the trajectory tracking of a tendon-driven BCM is successfully employed where the static manipulator equation is developed by using the principle of virtual work. Most of these model-based control approaches are computationally expensive. For faster numerical calculation, an inverse Cosserat-rod-based BCM

control is proposed [41]. The aforementioned model-based control approaches require accurate kinematic and dynamic formulations to control the task space of the BCMs. However, most of the developed control models to date suffer from a lack of accuracy due to a number of assumptions (constant curvature, omission of slackness in tendons, omission of gravitational effect and external load, structural deformation, hysteresis, uncertainties in kinematic and dynamic parameters, and so forth) considered for the mathematical formulation.

Model-free approaches are commonly used to control the conventional rigid arm robots [42], [43]. Using the concept for rigid arm robots [43], a low-level joint controller based on feed-forward NN is developed in [28] to control OCTARM VI soft robot. A combination of the NN and genetic algorithm is proposed to control a cooperative robot [44]. In [45], a fuzzy controller for kinematic control of continuum manipulator is developed, which skips evaluation of the Jacobian in each step. In one such control scheme reported in [46], the kinematic Jacobian matrix is estimated in real-time by steadily moving every actuator. A two-stage-task-space controller for compact bionic handling assistant is developed based on distal supervised learning and adaptive neural control [34]. In a similar effort, a recurrent NN is used to control a BCM in [47]. The above discussed model-free approaches offer much faster computation speed than the model-based approaches; however, those do not predict the inverse solutions accurately each time due to the nonlinearities of the BCM material behavior. Moreover, the model accuracy in a pneumatic actuated manipulator is way less than in tendon-driven robots due to the inherent material hysteresis nonlinearities and actuation delay. This technology gap leads to the requirement of an effective closed-loop controller for the end-effector trajectory of a pneumatic actuated continuum manipulator in the task space.

From the literature, it is found that the tip of a pneumatically actuated multi-segment BCM is very difficult to control due to the high flexibility, nonlinear material behavior, hysteresis, and memory effect [28]. The developed model-based controllers for the BCMs are mostly tracking controllers, which require an accurate dynamic model. Though the numerical inverse solution of some of the model-based methods is accurate, they are computationally complex and thus, unsuitable for real-time control applications [36]. To reduce the computational complexity and effects of nonlinearities and disturbances, etc., the use of a model-free approach appears to be a good choice. However, control architectures developed for rigid arm manipulators are unsuitable for pneumatically actuated BCMs. Thus, a specially tailored kinematic control architecture is required for pneumatically actuated continuum manipulators. This problem draws our attention to improve the BCM's trajectory tracking accuracy by using a kinematic controller in real-time.

In this regard, the major contributions of the present research are as follows.

- This paper proposes a combined model-free and model-based approach to exploit the advantages of the two methods.
- The developed hybrid real-time cascaded closed-loop kinematic control approach is able to deal with model ambiguity by controlling the JSPs of the pneumatically actuated BCMs.
- The proposed methodology uses a separate learning (SL) algorithm [48] governed NN model to find the intermediate segment pose from the desired tip pose of the manipulator and then again uses the NN model to find the arc parameters (arc length, arc angle, and arc radius of curvature) from the respective segment pose. The JSPs are then determined from the arc parameters by an inverse model of the piecewise constant curvature model to preserve the accuracy (geometric length constraints) of the actuator length [16].
- Since the obtained inverse solutions from the hybrid approach are not error-free, a cascaded closed-loop kinematic controller with a mid-level PI-controller is proposed. The developed controller controls the manipulator JSPs (actuator lengths) instead of the actuator space parameters (pneumatic actuation pressures) in order to eliminate the effect of material hysteresis and material memory effect. The required actuation pressures to attain the desired lengths are initially estimated by a NN model and are updated in each time step by model-free Jacobian regularization.
- The proposed hybrid kinematic control model is advantageous compared to the model-based approach in a way that it can accurately control the real-time trajectory tracking of a BCM by reducing the computational cost and thus, the feedback delay.
- Moreover, the neural network training uses position data at specific locations along the manipulator backbone, just like a high degree of freedom (DoF) manipulator. The training database size is drastically reduced by implementing the inverse piecewise constant curvature model in comparison to a purely model-free approach.
- The working behavior of the developed hybrid kinematic controller is applied to the Festo Didactics' trunk of the Robotino-XT (Fig. 1). Note that the kinematic controller neglects the dynamic (inertial) forces, and hence, it is valid for small accelerations.

The rest of this paper starts with a short description of Robotino-XT in Section-II. A generalized kinematic model for a serially connected BCM is described in Section-III by using NN and SL algorithms, and thereafter, the real-time control scheme is described in Section-IV. Section-V explains the experimental test bench (Robotino-XT and OptiTrack) and the workspace of the manipulator. Therein, the parameters used for the BCM control are given. Different test cases used to validate the proposed control algorithm and the corresponding experimental results are presented in Section VI. Finally, in Section-VII, conclusions and perspectives from this study are drawn.

II. ROBOTINO-XT

The Robotino-XT, schematically shown in Fig. 2, is a combination of a mobile Robotino and a portable bionic handling assistant (PBHA). The existing PBHA framework is connected to Robotino at its base and extended with a bio-mechatronic gripper at its head. The polyamide-made PBHA comprises two segments connected in series (See Fig. 2(a)). Each segment consists of three parallel-coupled individually actuated bellow tubes with 120° actuator spacing, as shown in Fig. 2(b). The 4.12° cross-sectional taper angle of the manipulator allows variable curvature bending of the segments. Each tube is pneumatically actuated by a compressor comprising two membrane pumps with a maximum design limit of 2 bar pressure. The inherently flexible structure of the bellow tubes is stiffened to a specific limit by pressurization. Note that the spatial displacement of the segment is predominantly due to two-axis bending, which results from differential pressure in the tubes of a segment. The axis extension of the backbone (a centerline connecting each section) is prevented by a steel wire (shown in Fig. 2(c)) attached to the backbone of the trunk. For each bellow tube, a wire-cable potentiometer measures its actual length at any state. These wire cables are routed through cable guides attached on the outside of each bellow tube.

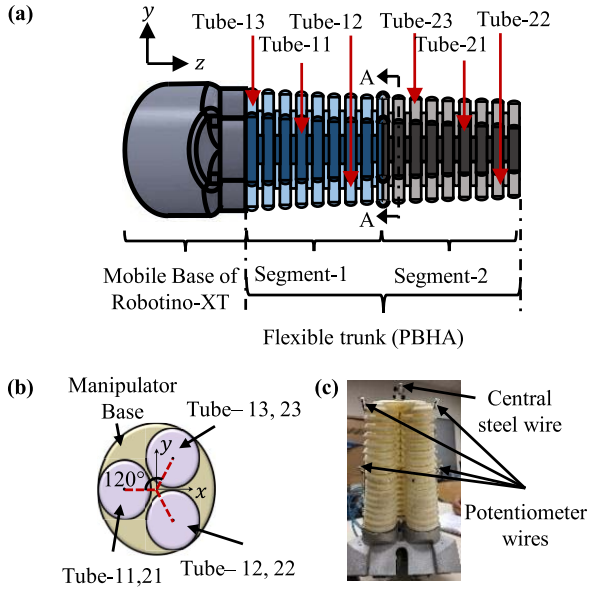

FIGURE 2. (a) Robotino-XT without gripper, (b) A-A sectional view of the PBHA, (c) The attached central steel wire in the manipulator backbone restricts axial movement, and the six potentiometer wire cables on the side walls of six actuators measure the tube lengths at any state. The control inputs are the separate pressures in the six tubes.

The distribution of pressures, manipulator self-weight, and applied external loading on the manipulator are shown by a free body diagram of segment **a**-**b** as in Fig. 3(a). It can be seen that the forces (F_P) due to uniform pressurization in the tubes act perpendicular to the cross-sectional plane. Note that in Fig. 3(a), the force due to fluid pressure is shown

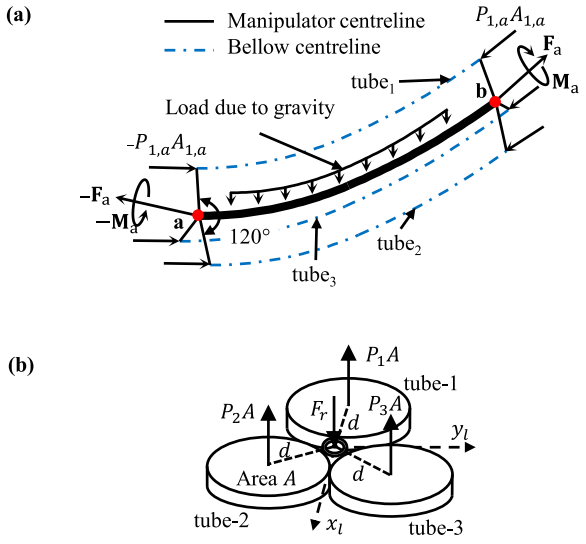

FIGURE 3. (a) Free body diagram of the manipulator segment a-b. The manipulator experiences pneumatic forces through bellow tubes, load due to gravity, and external loadings (b) Forces acting on the head (rigid hub) of the segment due to pneumatic actuation and steel rod.

only for bellow tube$_1$. By denoting the distal end area of the tube as A, the magnitude of the pneumatic force is PA. Since each segment of PBHA has three bellow tubes with 120° actuator spacing, the resultant moment created by the pressure generated forces (see Fig. 3(b)) is

$$M = \frac{\sqrt{3}\,dA}{2}(P_3 - P_2)\,\hat{i} + \frac{dA}{2}(2P_1 - P_2 - P_3)\,\hat{j},$$

where d is the distance from manipulator backbone to bellow tube backbone, and the resultant force is given as

$$P_1 + P_2 + P_3 - F_r = 0,$$

where F_r is the resisting force due to the central steel wire in the manipulator backbone. The inextensible steel wire restricts the axial extension of the manipulator and only allows a two-way bending. Note that the six potentiometer wires do not carry much tensile force. Apart from the pneumatic forces, the manipulator also experiences load due to gravity and externally applied loads. In Fig. 3(a), all the externally applied loads are lumped into the resultant wrench forces and moments (F, M) at points **a** and **b**.

The pneumatic actuators of PBHA have high material nonlinearity due to hysteresis in the polyamide material, mechanical friction, and pneumatic valves [34]. The aforementioned effects affect the pose of the tip to such an extent that the tip position is not unique for the same set of applied pressures from different initial tip positions (see Fig. 4). For example, Fig. 4 shows the tip position of the PBHA for four input pressure sets $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$ and $\mathbf{P}_4$; each applied ten times from different random initial tip positions. Note that each pressure set includes six tube pressures. Ideally, there should be a distinct tip position corresponding to each input pressure set and there should be four points in Fig. 4. However, it is

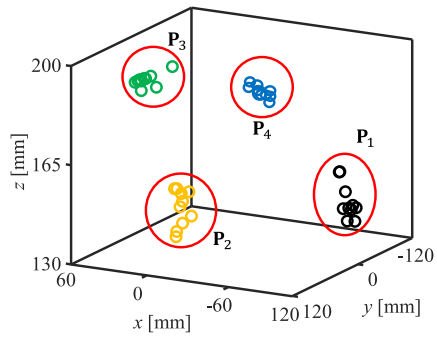

FIGURE 4. Illustration of tip position of the PBHA for input pressure sets: (P_1: [0.35,1.35,1.25,0.27,1.29,1.38] bar, P_2: [0.35,1.35,0.40,1.26,1.29,0.32] bar, P_3: [1.30,1.35,1.25,0.27,0.38,0.32] bar, P_4: [0.35,0.25,1.25,1.26,1.29, 0.32]) bar Here each pressure set is applied ten times from different random initial tip positions or pressures.

found that there are four clusters of points. This demonstrates how the material hysteresis, and dynamical uncertainties limit the positional control of the PBHA, i.e., the infeasibility of only pressure-based control. However, the tip position can be controlled accurately by controlling the tube lengths because each combination of the six tube lengths always gives a unique position. However, note that the converse is not necessarily true, i.e., for any given tip position, there can be none (if outside the workspace), one or more pressure combinations. Since the tube lengths are measured, those can be controlled by dynamically modifying the tube pressures. Thus, while the control objective is to position the robot tip somewhere and the control input is the tube pressures, the tube lengths can be introduced as intermediate control variables in the control loop. Note that an external system (OptiTrack) is used in this article to measure the robot tip position. However, OptiTrack measurements are not used in the control loop, and those are used here simply for independent validation of the experimental results.

Some possible backbone shapes of the PBHA in three dimensions are shown in Fig. 5(a), where frame $\{x_0, y_0, z_0\}$ is attached to the manipulator base, frame $\{x_1, y_1, z_1\}$ is attached to the base of the second segment, and frame $\{x_2, y_2, z_2\}$ is

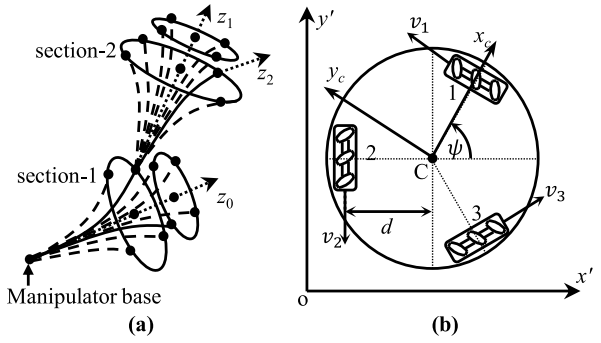

FIGURE 5. Modes of motion of (a) PBHA backbone, (b) Mobile base of Robotino-XT.

attached to the tip of the manipulator. Note that the curvilinear z-axis is aligned with the backbone shape, and bending occurs about local x and y axes perpendicular to the z-axis.

The 3-DoF mobile base of the Robotino-XT, as shown in Fig. 5(b), is fitted with a three-motor Omni-drive propulsion unit. The integrated infrared sensors measure the linear distance traversed in x and y-directions and the rotation about z-axis. The drive system and sensor module help the Robotino to follow a predefined path. The Robotino is equipped with a wireless real-time communication system to transmit the sensor data and address its drive units. The mobile base control, as pre-built in Robotino-XT, is extremely accurate, and hence, it is outside the scope of this article. Wherever the base movement will be utilized in this article, it would be assumed to be accurate, and any errors in tip position/trajectory would be assumed to be only due to the pneumatic actuated parts.

III. KINEMATIC MODEL

A. FORWARD KINEMATICS OF PBHA

This section focuses on the kinematic model of the PBHA of the Robotino-XT. The objective of the kinematic model is to find the position and orientation of the head coordinate frame from the actuator (bellow tube) lengths at any given state and vice versa. Here, the piecewise constant curvature model is used to find the pose of each PBHA segment from the length of the actuators. According to the piecewise constant curvature model, the manipulator segment shape is assumed to be a circular arc [14]. Since a circular arc can be uniquely defined by a set of arc parameters, the arc parameters of any PBHA segment can be defined as [16]

$$\mathfrak{k}_m = \begin{bmatrix} s_m & \beta_m & \kappa_m \end{bmatrix}^{\mathrm{T}}, \quad \mathfrak{k}_m \in \mathbb{R}^{3\times 1} \tag{1}$$

where $\mathfrak{k}$ is the vector of arc parameters, s is the arc length, β is the arc angle, κ is the curvature of the backbone of the segment, and $m \in 1, 2$ is the segment number. Note that, $(\cdot)^{\mathrm{T}}$ indicates the transpose of a matrix or vector. The arc parameters $\mathfrak{k}$ for m-th segment can be given in terms of the JSPs as [16]

$$\begin{aligned} s_m\left(l_{mn}\right) &= \left(l_{m1} + l_{m2} + l_{m3}\right)/3, \\ \beta_m\left(l_{mn}\right) &= \operatorname{atan2}\left(\sqrt{3}\left(l_{m3} - l_{m2}\right), l_{m2} + l_{m3} - 2l_{m1}\right) \\ \kappa_m\left(l_{mn}\right) &= \frac{2\sqrt{l_{m1}^2 + l_{m2}^2 + l_{m3}^2 - l_{m1}l_{m2} - l_{m2}l_{m3} - l_{m3}l_{m1}}}{\left(l_{m1} + l_{m2} + l_{m3}\right) r_{s_m}}, \end{aligned} \tag{2}$$

where r_s is the radius of the segment, l_{mn} is the length of the actuators, and $n \in 1, 2, 3$ is the tube number in each segment. Note that a single inextensible steel wire passes through the central backbone to connect the PBHA base to its tip. Thus, the length of each segment (s_1, s_2) may vary at different instances; however, the total length $(L = s_1 + s_2)$ of PBHA remains constant (see Fig. 2(c)). Note that some authors have wrongly considered s_1 and s_2 as constants [23], [34].

In Fig. 6(a), the head coordinate of a circular arc of radius r_c is given in xz-plane as $(r_c\left(1 - \cos\alpha\right), r_c \sin\alpha)$,

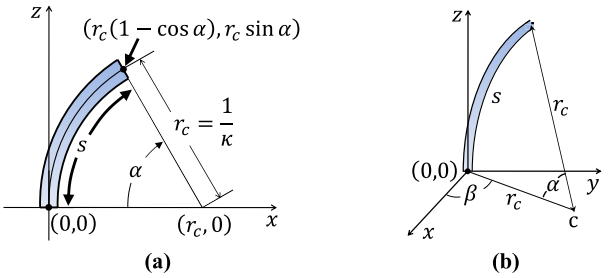

FIGURE 6. (a) Planar representation of a circular arc in *xz*-plane with arc parameters $\mathbb{t} = [s\ 0\ \kappa]^{\mathrm{T}}$, (b) Spatial representation of the circular arc with arc parameters $\mathbb{t} = [s\ \beta\ \kappa]^{\mathrm{T}}$.

i.e. $\mathbf{p} = [r_c(1-\cos\alpha)\ \ 0\ \ r_c\sin\alpha]^{\mathrm{T}}$, where $\alpha = \kappa s$. Here, it can be noted that the manipulator segment motion also comprises a rotation α about the positive y-axis. For spatial movement of the manipulator segment, the entire arc shown in Fig. 6(a) is rotated about the positive z-axis with an angle β as shown in Fig. 6(b). The double rotation produces a homogenous transformation $\mathbf{T} \in \mathbb{R}^{4\times4}$ from the base of the arc to the head of the arc. Thus, the transformation from the arc base to the tip is given as [15]

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}_z(\beta) & \mathbf{0}^{\mathrm{T}} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_y(\alpha) & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix}.$$

Noting that $r_c = 1/\kappa$, the transformation $\mathbf{T}$ from base to head of each manipulator segment in terms of arc parameters is given as

$$\mathbf{T}^{mb}_{m\mathfrak{h}} = \begin{bmatrix} c\alpha_m c\beta_m & -s\beta_m & s\alpha_m c\beta_m & \frac{c\beta_m(1-c\alpha_m)}{\kappa_m} \\ c\alpha_m s\beta_m & c\beta_m & s\alpha_m s\beta_m & \frac{s\beta_m(1-c\alpha_m)}{\kappa_m} \\ -s\alpha_m & 0 & c\alpha_m & \frac{s\alpha_m}{\kappa_m} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where $\alpha_m = \kappa_m s_m$, $\mathrm{c}(\cdot) \rightarrow \cos(\cdot)$, $\mathrm{s}(\cdot) \rightarrow \sin(\cdot)$, and $\left[\frac{c\beta_m(1-c\alpha_m)}{\kappa_m} \ \ \frac{s\beta_m(1-c\alpha_m)}{\kappa_m} \ \ \frac{s\alpha_m}{\kappa_m}\right]^{\mathrm{T}}$ is the Cartesian position of segment m. Using (3), $\mathbf{T}$ from world frame to tip of the first segment of the PBHA is obtained as

$$\mathbf{T}^{\mathrm{w}}_{1\mathrm{h}} = \mathbf{T}^{\mathrm{w}}_{1\mathrm{b}}\mathbf{T}^{1\mathrm{b}}_{1\mathrm{h}}, \quad \mathbf{T} \in \mathbb{R}^{4\times4} \tag{4}$$

Denoting $\mathbf{x}_1 = [x_1, y_1, z_1]^{\mathrm{T}}$ as the position of the head frame of the first segment from the world reference frame, and $\mathbf{x}'_2 = [x'_2, y'_2, z'_2]^{\mathrm{T}}$ as the position of the head frame of the second segment from the base of the second segment,

$$\begin{aligned} x_1 &= \mathbf{T}^{\mathrm{w}}_{1\mathrm{h}}(1,4), \\ y_1 &= \mathbf{T}^{\mathrm{w}}_{1\mathrm{h}}(2,4), \\ z_1 &= \mathbf{T}^{\mathrm{w}}_{1\mathrm{h}}(3,4), \\ x'_2 &= \mathbf{T}^{2\mathrm{b}}_{2\mathrm{h}}(1,4), \\ y'_2 &= \mathbf{T}^{2\mathrm{b}}_{2\mathrm{h}}(2,4), \end{aligned}$$

$$z'_2 = \mathbf{T}^{2\mathrm{b}}_{2\mathrm{h}}(3,4), \tag{5}$$

where $\mathbf{T}(i,j)$ represents i-th row and j-th column of $\mathbf{T}$. Note that the orientations are not independent variables in constant curvature model. In fact, orientations about x and y-axes are, respectively, found as $\theta_{x_1} = \mathrm{atan2}\left(-\mathbf{T}^{\mathrm{w}}_{1\mathrm{h}}(2,3), \mathbf{T}^{\mathrm{w}}_{1\mathrm{h}}(3,3)\right)$ and $\theta_{y_1} = \mathrm{atan2}\left(\mathbf{T}^{\mathrm{w}}_{1\mathrm{h}}(1,3), \mathbf{T}^{\mathrm{w}}_{1\mathrm{h}}(3,3)\right)$, where $\mathrm{atan2}(y,x) = \tan^{-1}(y/x)$ for $x > 0$, $\tan^{-1}(y/x) + \pi$ for $x < 0, y \geq 0$, $\tan^{-1}(y/x) - \pi$ for $x < 0, y < 0$, $(\pi\,\mathrm{sign}(y))/2$ for $x = 0$, and undefined for $x = 0, y = 0$. Orientation about the z-axis is computed from $\cos^2\theta_{z_1} = 1 - cos^2\theta_{x_1} - \cos^2\theta_{y_1}$.

Likewise, using (3), $\mathbf{T}$ from world frame to tip of the PBHA is obtained as

$$\mathbf{T}^{\mathrm{w}}_{2\mathrm{h}} = \mathbf{T}^{\mathrm{w}}_{1\mathrm{b}}\prod_{m=1}^{2}\mathbf{T}^{mb}_{m\mathfrak{h}}\mathbf{T}^{m\mathfrak{h}}_{(m+1)\mathrm{b}}, \quad \mathbf{T} \in \mathbb{R}^{4\times4} \tag{6}$$

where the transformation from m-th segment head to $(m+1)$-th segment base is an identity matrix because the thin hub between any two segments is rigid.

The position of the second PBHA segment is $\mathbf{x}_2 = [x_2, y_2, z_2]^{\mathrm{T}}$, where

$$\begin{aligned} x_2 &= \mathbf{T}^{\mathrm{w}}_{2\mathrm{h}}(1,4), \\ y_2 &= \mathbf{T}^{\mathrm{w}}_{2\mathrm{h}}(2,4), \\ z_2 &= \mathbf{T}^{\mathrm{w}}_{2\mathrm{h}}(3,4). \end{aligned} \tag{7}$$

Note that $\mathbf{x}_2 = \boldsymbol{f}\left(\mathbf{x}_1, \mathbf{x}'_2\right)$. As per constant curvature model, the orientations are $\theta_{x_2} = \mathrm{atan2}\left(-\mathbf{T}^{\mathrm{w}}_{2\mathrm{h}}(2,3), \mathbf{T}^{\mathrm{w}}_{2\mathrm{h}}(3,3)\right)$ and $\theta_{y_2} = \mathrm{atan2}\left(\mathbf{T}^{\mathrm{w}}_{2\mathrm{h}}(1,3), \mathbf{T}^{\mathrm{w}}_{2\mathrm{h}}(3,3)\right)$. Orientation about z-axis is computed from $\cos^2\theta_{z_2} = 1 - \cos^2\theta_{x_2} - \cos^2\theta_{y_2}$.

The FK schematic of the PBHA by using a piecewise constant curvature model and the manipulator base as the world frame is shown in Fig. 7. It can be observed that $\mathbf{x}_1$ is a function of l_{11}, l_{12} and l_{13}, and $\mathbf{x}'_2$ is a function of l_{21}, l_{22} and l_{23}; whereas the pose of the second PBHA segment $\mathbf{x}_2$ is a function of $l_{11}, l_{12}, l_{13}, l_{21}, l_{22}$ and l_{23}.

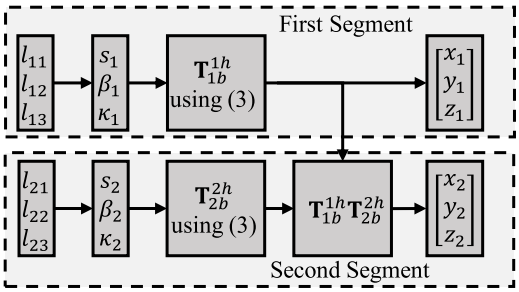

FIGURE 7. FK model of each segment of the PBHA depicting the mapping from the JSPs to TSPs.

B. INVERSE KINEMATICS OF PBHA

Since the PBHA is an under-actuated manipulator, multiple backbone postures are possible for a particular desired tip position. For example, consider a special planar case with two segments, $\beta_1 = \beta_2 = 0$, and $\alpha_2 = -\alpha_1$ (See Fig. 6). Then the tip of the second segment has position

$\mathbf{x}_2 = \left[(s_1 + s_2)\,(1 - \cos(\alpha_1))\,/\alpha_1, 0, (s_1 + s_2)\sin(\alpha_1)\,/\alpha_1\right]^{\mathrm{T}}$ and orientation $\boldsymbol{\theta}_2 = \left[\pi/2 \;\; \pi/2 \;\; 0\right]$ rad, i.e., along the z-axis. Keeping the angles $\alpha_1, \alpha_2, \beta_1$ and $\beta_2$fixed, the position and orientation of the tip of the second segment do not change if $L = s_1 + s_2$ is constant. For this example, there is an infinite number of combinations of values of s_1 and s_2 that produce the same position and orientation of the tip of the second segment. This demonstrates that the inverse solution of a PBHA is not unique for a given tip position. In fact, determination of the optimal inverse solutions of PBHA is quite complicated [36]. Moreover, it is challenging to deduce the IK resolutions from the model-based FK models because of the complicated nonlinear mathematical formulations involved in the 3D geometry solution. For example, the pose of the first segment ($\mathbf{x}_1$) cannot be uniquely determined from the tip pose ($\mathbf{x}_2$), which in turn hampers the determination of the inverse solution, i.e., l_{11}, l_{12}, l_{13}, l_{21}, l_{22} and l_{23}. Besides, several assumptions made during the development of the model-based approaches limit the accuracy of IK.

Thus, this paper introduces a hybrid method based on both model-based and model-free techniques to find the inverse solution (schematically represented in Fig. 8) of the PBHA to avoid the above-discussed problems. The hybrid method uses penalty parameters in the model-free step so that for a given initial tip pose and segment backbone shapes, the neighborhood pose on the robot tip trajectory is optimally selected from many possible solutions. Thereafter, using the model-based piecewise constant curvature technique, the lengths of the actuators from the arc parameters of the corresponding segment are obtained by inverting (2) as $l_{mn} = g^{-1}(\mathfrak{k}_m)$.

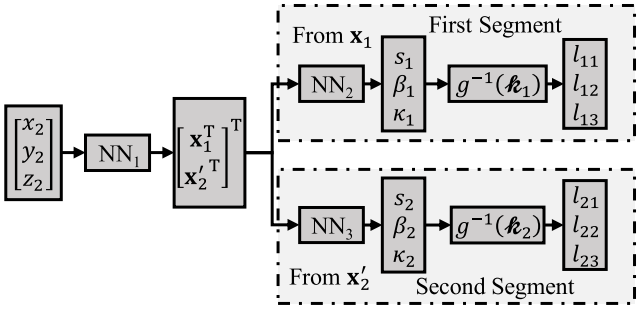

FIGURE 8. Hybrid IK Model of PBHA where NN_1 is used to find each segment's head frame from their base frame, and NN_2, NN_3 are used to find arc parameters of first and second segment respectively.

The hybrid inverse model consists of a model-free technique in the form of NN to find the $\mathbf{x}_1$ and $\mathbf{x}_2'$ directly from $\mathbf{x}_2$ in the case of PBHA. Additionally, two more NN models are used to find the arc parameters $\mathfrak{k}_1$ and $\mathfrak{k}_2$ from $\mathbf{x}_1$ and $\mathbf{x}_2'$, respectively. Thus, for m segment manipulator, one NN model is required to find the base frame to head frame pose of each segment directly from the position of the PBHA tip. Moreover, additional m number of NN models are required to find the corresponding arc parameters of the segments. Therefore, for m segment manipulator, $(m+1)$ number of NN models are required to solve the IK with the proposed hybrid approach.

This paper considers a three-layer NN model having an input layer (I), a hidden layer (H), and an output layer (O). The output $\mathbf{r}$ of the NN model is given as [43]

$$\mathbf{r} = f_2\left(\mathbf{W}_2{}^{\mathrm{T}} f_1\left(\mathbf{W}_1{}^{\mathrm{T}}\mathbf{p}\right)\right). \tag{8}$$

where $\mathbf{p}$ is the given network input, $\mathbf{W}_1$ is the weight matrix between the input (I) and hidden (H) layers, and $\mathbf{W}_2$ is the weight matrix between the hidden (H) and output (O) layers, and $f_1(\cdot)$ and $f_2(\cdot)$ are the activation functions.

The SL algorithm for NN training is a supervised learning process considered here to update the weight values of the network [48]. Since both over-training and under-training affect the accuracy and stability of the network, the SL algorithm attempts to eliminate over-fitting and under-fitting of the network. Here, the SL algorithm is implemented in the three-layer network, which is divided into two sub networks. A loss function $L(\mathrm{T} - \mathbf{r})$ is defined, which measures the difference between actual output T and predicted output $\mathbf{r}$. For the first sub-network i.e., hidden layer to output layer, the empirical risk $E_{\mathrm{H\text{–}O}}[\mathbf{W}_2]$ for the averaged samples is modified and defined as

$$\begin{aligned} E_{\mathrm{H\text{–}O}}[\mathbf{W}_2] = {} & \frac{1}{2}\sum_l \sum_{k=1}^{N_3}\left[\mathrm{T}_k^l - \mathbf{r}_k^l\right]^2 \\ & + \frac{1}{2}\lambda\left(\|\mathbf{W}_2\|_2^2 + \Delta\Omega\right). \\ & \text{s.t. } \Omega_{\min} \le \Omega \le \Omega_{\max} \end{aligned} \tag{9}$$

where $\mathrm{T} \in \mathbb{R}^{N_3 \times 1}$ is the target vector, l is the number of training scenarios, $\mathbf{r}_k^l$ is the output to the kth output unit for the l-th training scenario, λ is the penalty term, $\|\cdot\|_2$ is the Euclidian norm, Ω is the penalty regularization term, and $\Omega_{\min}$, and $\Omega_{\max}$ are minimum and maximum value of Ω respectively. The penalty regularization term is defined separately for different neural networks. For instance, Ω is defined as

$$\Delta\Omega = \begin{cases} \|\Delta\mathbf{x}_1\|_2^2 + \left\|\Delta\mathbf{x}_2'\right\|_2^2, & \text{for } NN_1 \\ \|\Delta\mathfrak{k}\|_2^2, & \text{for } NN_2 \text{ and } NN_3. \end{cases}$$

For NN_1, additional constraints are defined as

$$\mathbf{x}_{1\min} \le \mathbf{x}_1 \le \mathbf{x}_{1\max}, \quad \mathbf{x}'_{2\min} \le \mathbf{x}_2' \le \mathbf{x}'_{2\max},$$

and for NN_2, an additional constraint is defined as

$$\mathfrak{k}_{\min} \le \mathfrak{k} \le \mathfrak{k}_{\max}.$$

The penalty term and the squared network output are added to the empirical risk function to select a particular solution from multiple possible solutions. The use of the penalty term in (9) offers an effective way to control the magnitude of squared network output, and thus, a particular inverse solution of a network is selected. Moreover, the penalty due to the additional constraints allows the network outputs to lie within a prescribed range. For instance, in the case of NN_3 in Fig. 8, the arc parameters ($\left[s_1 \;\; \beta_1 \;\; \kappa_1\right]^{\mathrm{T}}$) are considered as a penalty

to the network such that unique arc parameters are chosen. In literature, it has been established that the squared network output is small for a larger penalty term [49]. In the case of PBHA, the Euclidian norm of the empirical risk function in (9) is minimized for a good performance in terms of MSE for a penalty term value of $\lambda = 0.001$.

The weight is updated by using the damped-least square method and Newton's method of approximation. Furthermore, by using the updated weight values, the ideal value of the hidden layer is determined [48]. Determination of the ideal values of the hidden layer completes the first training process of NN by using the SL algorithm. Similarly, the derived ideal values for l-th training scenario are further used as a target vector for the second sub-network training between the input and hidden layers.

The SL algorithm effectively trains the NN based on the 'Divide-and-conquer method', which does not use random hidden output values or perturbation parameters. Here, the three-layer network is divided into two sub-networks as I-H and H-O, which are trained separately. The training of H-O is similar to back propagation algorithm. However, to train the I-H, the ideal values of H are determined separately by using Newton's approximation method.

The proposed inverse hybrid model for PBHA is a combination of both a piecewise constant curvature model (model-based) and a NN (model-free). The model avoids the complex mathematical nonlinear equations and gives a better alternative for fast and accurate optimal solutions for the real-time control of the BCM.

IV. KINEMATIC CONTROL OF BCM

A. KINEMATIC SATURATION

To design an efficient controller that allows the PHBA to respond correctly to commands, the actuation pressures (in terms of gauge values) must be within acceptable ranges. As per the manufacturer of the PBHA, the maximum allowable pressure for the bellow tubes are in the range of $P_{mn} \in \begin{bmatrix} 0 & 2 \end{bmatrix}$ bar. If all the actuation pressures P_{mn} lie in the allowable range, then the low-level controller input is equal to the controller output. However, if any actuation pressure P_{mn} does not lie in the allowable range, then the low-level controller modifies the input pressures to avoid saturation. If any $P_{mn} < 0$ bar then the particular input command is converted to 0 bar, i.e. $P_{mn} = 0$. However, if one or more $P_{mn} > 2$ bar, then the maximum value is determined from the lot as

$$\begin{aligned} P_{\max} &= \max\left(\widehat{\mathbf{P}}\right) \\ &= \max\left(\begin{bmatrix} \widehat{P}_{11}, & \widehat{P}_{12}, & \widehat{P}_{13}, & \widehat{P}_{21}, & \widehat{P}_{22}, & \widehat{P}_{23} \end{bmatrix}^{\mathrm{T}}\right) \end{aligned} \tag{10}$$

where $\widehat{\mathbf{P}} \in \mathbb{R}^{6\times 1}$ is the updated pressure set from the controller. To get the controller output within the defined pressure range, normalization of $\widehat{\mathbf{P}}$ is done. Therefore, the pneumatic tube pressures are scaled to

$$\mathbf{P} = \begin{cases} \widehat{\mathbf{P}}, & \forall\,(m,n) : P_{mn} \leq 2\,\text{bar} \\ \dfrac{P_{\text{allowable}}}{P_{\max}}\widehat{\mathbf{P}}, & \exists\,(m,n) : P_{mn} > 2\,\text{bar}. \end{cases} \tag{11}$$

The input actuation pressures to the manipulator through the low-level controller are updated by using (11). This low-level controller is in-built with Robotino-XT and cannot be modified or over-ridden.

B. CONTROLLER MODELLING

The PBHA is actuated by a filtered input pressure set $\mathbf{P}$ through a low-level controller. This low-level controller comes preinstalled in the Robotino-XT setup, and it cannot be modified. Therefore, any modified control architecture for the tip pose accuracy is constrained to use this preinstalled controller. The in-built controller in the Robotino-XT is an open-loop controller and has been so far used by researchers for only position control [36].

Thus, in this section, an efficient closed-loop kinematic controller for a pneumatic actuated multi-segment BCM (here, for PBHA) is proposed to control the tip pose by controlling the length of the actuators, and finally, extended to a trajectory tracking problem. It is noted that the path control problem will be treated here as a piece-wise (or step-through) position control problem due to the inherent slow response of the pneumatic actuation system.

For the desired position $\boldsymbol{\chi}_d$, the hybrid IK model, as discussed in Section-III(B) is used to determine the desired JSPs of the PBHA. Moreover, the NN model is also used to find an initial nominal actuation pressure set $\mathbf{P}_0$ from the desired length of the actuators. Since multiple solutions may exist for the desired tip pose, a penalty regularization is performed to select an optimal actuation pressure set. This paper selects the regularization term (Euclidian norm) by assuming the minimum pressure variation required to reach a target position. Since the volumetric change in each bellow is negligible due to bending of the bellow, the actuation work under isothermal conditions is primarily due to pressure change, and the energy consumption is indirectly minimized.

Here, a PI-controller is used to control the length of the actuators, where the difference in the desired actuator lengths $\mathbf{l}_d$ and the actual actuator lengths $\mathbf{l}_a$ of the bellow tubes are the inputs to the controller. Assuming that the output of the controller is the pressure difference $\Delta\widehat{\mathbf{P}}(t)$, the pneumatic pressures for actuation is updated by

$$\widehat{\mathbf{P}}(t) = \widehat{\mathbf{P}}(t-1) + \Delta\widehat{\mathbf{P}}(t) \tag{12}$$

where $\widehat{\mathbf{P}}(t)$ is the actuation pressures in the current state, and t is the current step. The change in pressure required for length correction is given as

$$\Delta\widehat{\mathbf{P}}(t) \triangleq \mathbf{K_P}\mathbf{e}(t) + \mathbf{K_I}\mathbf{e_I}(t), \quad \Delta\widehat{\mathbf{P}} \in \mathbb{R}^{6\times 1} \tag{13}$$

where t is the current step, $\mathbf{K_P}$ is the proportional gain matrix and $\mathbf{K_I}$ is the integral gain matrix. The error $\mathbf{e}(t)$ and its

discrete-time integral $\mathbf{e}_{\mathrm{I}}(t)$ are defined as (14) and (15), shown at the bottom of the page.

The PI-controller gain parameters $\mathbf{K}_{\mathrm{P}}$ and $\mathbf{K}_{\mathrm{I}}$ are tuned by minimizing the total mean square error (MSE) of the system. Therefore, total MSE is defined by the total cost function as

$$E = E_{\mathrm{P}} + E_{\mathrm{I}} \tag{16}$$

where E_{P} is the mean cost function for proportional constant and is defined as

$$E_{\mathrm{P}} = 0.5 \sum_{\mathfrak{h}=0}^{\infty} \mathbf{e}^{\mathrm{T}}(t+\mathfrak{h})\, \mathbf{W}_{\mathrm{P}} \mathbf{e}(t+\mathfrak{h}), \tag{17}$$

and E_{I} is the mean cost function for integral constant and is defined as

$$E_{\mathrm{I}} = 0.5 \sum_{\mathfrak{h}=0}^{\infty} \mathbf{e}_{\mathrm{I}}^{\mathrm{T}}(t+\mathfrak{h})\, \mathbf{W}_{\mathrm{I}} \mathbf{e}_{\mathrm{I}}(t+\mathfrak{h}). \tag{18}$$

where $\mathbf{W}_{\mathrm{P}} \in \mathbb{R}^{6\times 6}$ and $\mathbf{W}_{\mathrm{I}} \in \mathbb{R}^{6\times 6}$ are positive definite diagonal weight matrices.

To minimize the total cost function E in (16), a learning rule is implemented. Therein, the change in proportional gain parameters are defined as

$$\Delta \mathbf{K}_{\mathrm{P}} = -\eta \left(\partial E \big/ \partial \mathbf{K}_{\mathrm{P}} \right) \tag{19}$$

where η is the positive definite learning rate, and its value should be chosen to be sufficiently small.

If it is assumed that the current change in input pneumatic pressure $\mathbf{P}(t)$ largely affects up to $\mathfrak{n}$-th future iteration step actuator length $\mathbf{l}_a(t+\mathfrak{n})$, then $\Delta \mathbf{K}_{\mathrm{P}}$ in (19) can be rewritten by using the chain rule as

$$\begin{aligned} \Delta \mathbf{K}_{\mathrm{P}} &= -\eta \left[\frac{\partial E}{\partial E_{\mathrm{P}}} \frac{\partial E_P}{\partial \mathbf{e}(t+\mathfrak{n})} \frac{\partial \mathbf{e}(t+\mathfrak{n})}{\partial \mathbf{l}_a(t+\mathfrak{n})} \frac{\partial \mathbf{l}_a(t+\mathfrak{n})}{\partial \mathbf{P}(t)} \frac{\partial \mathbf{P}(t)}{\partial \Delta \widehat{\mathbf{P}}(t)} \right]^{\mathrm{T}} \\ &\quad \times \frac{\partial \Delta \widehat{\mathbf{P}}(t)}{\partial \mathbf{K}_{\mathrm{P}}}, \end{aligned} \tag{20}$$

where

$$\begin{aligned} \frac{\partial E}{\partial E_{\mathrm{P}}} &= 1, \\ \frac{\partial E_{\mathrm{P}}}{\partial \mathbf{e}(t+\mathfrak{n})} &= \mathbf{W}_{\mathrm{P}} \mathbf{e}(t+\mathfrak{n}), \\ \frac{\partial \mathbf{e}(t+\mathfrak{n})}{\partial \mathbf{l}_a(t+\mathfrak{n})} &= -\mathbf{I}, \\ \frac{\partial \mathbf{l}_a(t+\mathfrak{n})}{\partial \mathbf{P}(t)} &= \mathbf{J}, \\ \frac{\partial \mathbf{P}(t)}{\partial \Delta \widehat{\mathbf{P}}(t)} &= \varepsilon = 1 \text{ or } \frac{P_{\text{allowable}}}{\mathrm{P}_{\max}}, \\ \frac{\partial \Delta \widehat{\mathbf{P}}(t)}{\partial \mathbf{K}_{\mathrm{P}}} &= \mathbf{e}^{\mathrm{T}}(t), \end{aligned}$$

$\mathbf{I}$ is 6×6 identity matrix, and $\mathbf{J}$ is the Jacobian matrix. The Jacobian $\mathbf{J} \in \mathbb{R}^{mn\times op}$ is defined as [50]

$$\begin{aligned} \mathbf{J} &= \frac{\partial \mathbf{l}_a(t+\mathfrak{n})}{\partial \mathbf{P}(t)} \\ &= \begin{bmatrix} \frac{\partial \mathrm{l}_{11,\mathrm{a}}(t+\mathfrak{n})}{\partial \mathrm{P}_{11}(t)} & \cdots & \frac{\partial \mathrm{l}_{11,\mathrm{a}}(t+\mathfrak{n})}{\partial \mathrm{P}_{op}(t)} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathrm{l}_{mn,\mathrm{a}}(t+\mathfrak{n})}{\partial \mathrm{P}_{11}(t)} & \cdots & \frac{\partial \mathrm{l}_{mn,\mathrm{a}}(t+\mathfrak{n})}{\partial \mathrm{P}_{op}(t)} \end{bmatrix}, \end{aligned} \tag{21}$$

where l_{a} is the actual length of the actuator tube, subscripts $m, o = 1, 2$ refer to the segment number, and $n, p = 1, 2, 3$ refer to the bellow tube number. The PI controller gains are tuned by using the Jacobian from an offline trained NN model for model linearization, i.e., for computing $\frac{\partial \mathrm{l}_{\mathrm{mn}}}{\partial \mathrm{P}_{\mathrm{op}}}$. The Jacobian is a block diagonal matrix with two 3×3 off-diagonal null matrices because actuator lengths in the bellows in a segment do not change when any bellow of the other segment is pressurized, i.e., $\frac{\partial \mathrm{l}_{\mathrm{mn,a}}}{\partial \mathrm{P}_{\mathrm{op}}} = 0$ when $m \neq o$. Inserting all the partial derivatives and rearranging, (20) can be written as

$$\Delta \mathbf{K}_{\mathrm{P}} = \eta \varepsilon \mathbf{J}^{\mathrm{T}} \mathbf{W}_{\mathrm{P}} \mathbf{e}(t+\mathfrak{n})\, \mathbf{e}^{\mathrm{T}}(t). \tag{22}$$

Similar to (22), the change in integral gain parameters $\Delta \mathbf{K}_{\mathrm{I}}$ is defined as

$$\Delta \mathbf{K}_{\mathrm{I}} = \eta \varepsilon \mathbf{J}^{\mathrm{T}} \mathbf{W}_{\mathrm{I}} \mathbf{e}_{\mathrm{I}}(t+\mathfrak{n})\, \mathbf{e}_{\mathrm{I}}^{\mathrm{T}}(t). \tag{23}$$

Note that the integral control is only used to reduce steady-state error. If the integral control is assumed to start from $t_1 \geq 0$ then

$$\Delta \mathbf{W}_{\mathrm{I}} = 0, \quad t < t_1. \tag{24}$$

Using (22), and (23), the gain matrices are updated for task space control of the BCM. The proposed tuning process is repeated until the control gain parameters $\Delta \mathbf{K}_{\mathrm{P}}$ and $\Delta \mathbf{K}_{\mathrm{I}}$ nearly approach zero and does not reduce appreciably thereafter, which also indicates a fully minimized cost function defined in (16).

A block diagram of the proposed closed-loop cascaded PI-controller is shown in Fig. 9. Since it has been assumed for the controller that the current input pneumatic pressure $\mathbf{P}(t)$ largely affects up to $\mathfrak{n}$-th future step output actuator length $\mathbf{l}_a(t+\mathfrak{n})$, the modeling shows that the $\mathbf{K}_{\mathrm{P}}$ and $\mathbf{K}_{\mathrm{I}}$ are updated by an NN-based Jacobian matrix for n step future error.

Note that the external visual processing system (OptiTrack), as in Fig. 9, does not participate actively in controlling the tip of the PBHA. It is used only as an observer to record the position and trajectory in real-time to determine the performance and accuracy of the developed controller.

$$\mathbf{e}(t) = \Delta \mathbf{l}(t) = \mathbf{l}_d(t) - \mathbf{l}_a(t) \tag{14}$$

$$\mathbf{e}_{\mathrm{I}}(t) = \sum \mathbf{e}(t) = \left[\sum_{\varsigma=1}^{t} \mathrm{e}_{11}(\varsigma), \quad \sum_{\varsigma=1}^{t} \mathrm{e}_{12}(\varsigma), \quad \cdots, \quad \sum_{\varsigma=1}^{t} \mathrm{e}_{23}(\varsigma) \right]^{\mathrm{T}}. \tag{15}$$

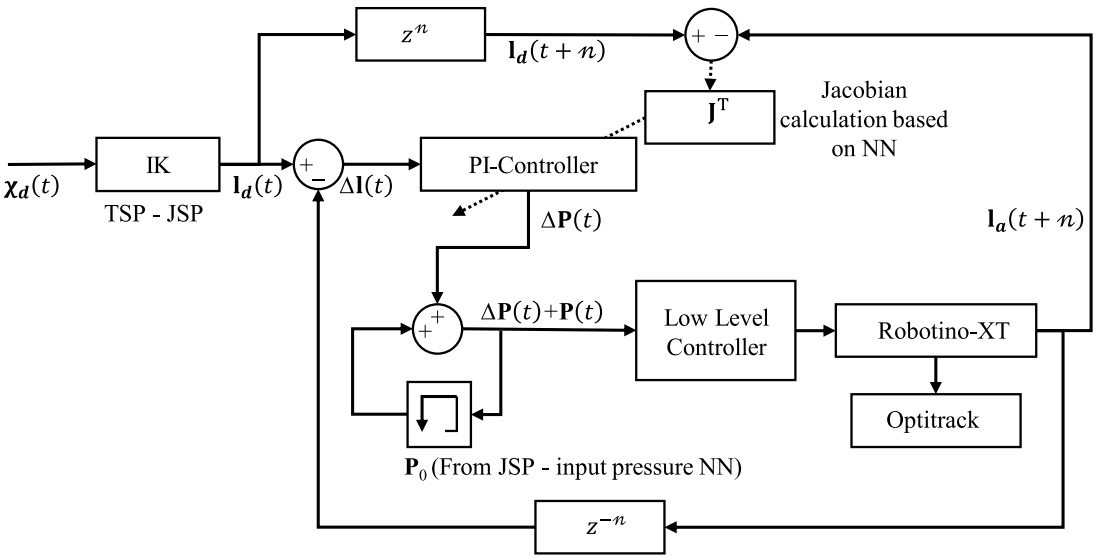

FIGURE 9. Closed-loop cascaded control architecture for real-time implementation.

C. CONVERGENGE OF THE CONTROLLER

For the proposed control scheme defined in (22) and (23), it is assumed that the command steps between the adjacent points on the end effector trajectory are sufficiently small such that the evolution of states is locally Lipschitz and linear. Then let the linearized state equation for the PBHA in discrete-time be defined as

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{P}(t), \tag{25a}$$
$$\mathbf{l}_a(t+1) = \mathbf{C}\mathbf{x}(t+1), \tag{25b}$$

where $\mathbf{x}$ is the state vector, and $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are the state, input, and output matrices, respectively. If it is assumed that the current $\mathbf{P}(t)$ affects up to $\mathfrak{n}$-th future output actuator length $\mathbf{l}_a(t+\mathfrak{n})$, then the Jacobian matrix is given by using the chain rule as

$$\mathbf{J} = \frac{\partial \mathbf{l}_a(t+\mathfrak{n})}{\partial \mathbf{P}(t)} = \frac{\partial \mathbf{l}_a(t+\mathfrak{n})}{\partial \mathbf{x}(t+\mathfrak{n})}\frac{\partial \mathbf{x}(t+\mathfrak{n})}{\partial \mathbf{P}(t)} = \mathbf{C}\mathbf{A}^{\mathrm{n-1}}\mathbf{B} \tag{26}$$

The state and measurement updates are separated. It is assumed that before the parameter update process begins, the length of actuators are given as $\mathbf{l}_{a_0}(t+1)$ for an input $\mathbf{P}_0(t)$. At this step, the state equation is written as

$$\mathbf{x}_0(t+1) = \mathbf{A}\mathbf{x}_0(t) + \mathbf{B}\mathbf{P}_0(t)$$
$$\mathbf{l}_{a_0}(t+1) = \mathbf{C}\mathbf{x}_0(t+1) \tag{27}$$

The gain parameters $\mathbf{K}_{\mathrm{P}}$ and $\mathbf{K}_{\mathrm{I}}$ need to be updated if the output error after $\mathfrak{n}$ number of future steps does not converge to zero. Then the input pressure $\mathbf{P}_0(t)$ is adjusted by updating $\Delta\mathbf{P}_0(t)$ as

$$\Delta\mathbf{P}_0(t) = \Delta\mathbf{K}_{\mathrm{P}}\mathbf{e}_0(t) + \Delta\mathbf{K}_{\mathrm{I}}\mathbf{e}_{\mathrm{I},0}(t). \tag{28}$$

The input pressure update $\Delta\mathbf{P}_0(t)$ in (28) can be rewritten for $\mathfrak{n}$ number of future steps by inserting $\Delta\mathbf{K}_{\mathrm{P}}$ and $\Delta\mathbf{K}_{\mathrm{I}}$ from (22) and (23) as

$$\Delta\mathbf{P}_0(t) = \eta\varepsilon\Big[\mathbf{J}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\mathbf{e}_0(t+\mathfrak{n})\,\mathbf{e}_0^{\mathrm{T}}(t)\,\mathbf{e}_0(t) + \mathbf{J}^{\mathrm{T}}\mathbf{W}_{\mathrm{I}}\mathbf{e}_{\mathrm{I},0}(t+\mathfrak{n})\,\mathbf{e}_{\mathrm{I},0}^{\mathrm{T}}(t)\,\mathbf{e}_{\mathrm{I},0}(t)\Big] \tag{29}$$

Noting that $\mathbf{e}_0^{\mathrm{T}}(t)\,\mathbf{e}_0(t) \geq 0$, $\mathbf{e}_{\mathrm{I},0}^{\mathrm{T}}(t)\,\mathbf{e}_{\mathrm{I},0}(t) \geq 0$ and $\eta \geq 0$, define two variables $\eta_1 = \eta\mathbf{e}_0^{\mathrm{T}}(t)\,\mathbf{e}_0(t) \geq 0$ and $\eta_2 = \eta\mathbf{e}_{\mathrm{I},0}^{\mathrm{T}}(t)\,\mathbf{e}_{\mathrm{I},0}(t) \geq 0$. Then (29) can written as

$$\Delta\mathbf{P}_0(t) = \varepsilon\Big[\eta_1\mathbf{J}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\mathbf{e}_0(t+\mathfrak{n}) + \eta_2\mathbf{J}^{\mathrm{T}}\mathbf{W}_{\mathrm{I}}\mathbf{e}_{\mathrm{I},0}(t+\mathfrak{n})\Big]. \tag{30}$$

The updated pneumatic pressure is then written as

$$\mathbf{P}_1(t) = \mathbf{P}_0(t) + \Delta\mathbf{P}_0(t). \tag{31}$$

The subsequent state update is performed by using the updated pneumatic pressures as

$$\mathbf{x}_1(t+1) = \mathbf{A}\mathbf{x}_0(t) + \mathbf{B}\mathbf{P}_1(t). \tag{32}$$

Equation (32) can be rewritten by inserting (31) as

$$\mathbf{x}_1(t+1) = \mathbf{A}\mathbf{x}_0(t) + \mathbf{B}\left(\mathbf{P}_0(t) + \Delta\mathbf{P}_0(t)\right). \tag{33}$$

Using $\mathbf{x}_0(t+1) = \mathbf{A}\mathbf{x}_0(t) + \mathbf{B}\mathbf{P}_0(t)$ from (24a) and inserting (29) into (33), $\mathbf{x}_1(t+1)$ is given as

$$\mathbf{x}_1(t+1) = \mathbf{x}_0(t+1) + \varepsilon\mathbf{B}\Big[\eta_1\mathbf{J}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\mathbf{e}_0(t+\mathfrak{n})\,\mathbf{e}_0^{\mathrm{T}}(t)\,\mathbf{e}_0(t) + \eta_2\mathbf{J}^{\mathrm{T}}\mathbf{W}_{\mathrm{I}}\mathbf{e}_{\mathrm{I},0}(t+\mathfrak{n})\,\mathbf{e}_{\mathrm{I},0}^{\mathrm{T}}(t)\,\mathbf{e}_{\mathrm{I},0}(t)\Big] \tag{34}$$

In a similar manner, the $\mathfrak{n}$ step future state $\mathbf{x}_1(t+\mathfrak{n})$ is written as

$$\mathbf{x}_1(t+\mathfrak{n}) = \mathbf{A}\mathbf{x}_1(t+\mathfrak{n}-1) + \mathbf{B}\mathbf{P}_1(t+\mathfrak{n}-1)$$
$$= \mathbf{A}^{\mathrm{n-1}}\Big[\mathbf{x}_0(t+1)$$

$$+\varepsilon\mathbf{B}\left(\eta_1\mathbf{J}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\mathbf{e}_0\left(t+\mathfrak{n}\right)+\eta_2\mathbf{J}^{\mathrm{T}}\mathbf{W}_{\mathrm{I}}\mathbf{e}_{\mathrm{I},0}\left(t+\mathfrak{n}\right)\right)\Big]$$
$$=\mathbf{x}_0\left(t+\mathfrak{n}\right)+\varepsilon\mathbf{A}^{\mathrm{n}-1}\mathbf{B}\Big[\eta_1\mathbf{J}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\mathbf{e}_0\left(t+\mathfrak{n}\right)+\eta_2\mathbf{J}^{\mathrm{T}}\mathbf{W}_{\mathrm{I}}\mathbf{e}_{\mathrm{I},0}\left(t+\mathfrak{n}\right)\Big], \tag{35}$$

and the output actuator length is written as

$$\begin{aligned}\mathbf{l}_{a,1}\left(t+\mathfrak{n}\right)&=\mathbf{C}\mathbf{x}_1\left(t+\mathfrak{n}\right)\\&=\mathbf{C}\mathbf{x}_0\left(t+\mathfrak{n}\right)+\varepsilon\mathbf{C}\mathbf{A}^{\mathrm{n}-1}\mathbf{B}\\&\quad\times\Big[\eta_1\mathbf{J}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\mathbf{e}_0\left(t+\mathfrak{n}\right)+\eta_2\mathbf{J}^{\mathrm{T}}\mathbf{W}_{\mathrm{I}}\mathbf{e}_{\mathrm{I},0}\left(t+\mathfrak{n}\right)\Big]\\&=\mathbf{l}_{a,0}\left(t+\mathfrak{n}\right)+\varepsilon\mathbf{C}\mathbf{A}^{\mathrm{n}-1}\mathbf{B}\\&\quad\times\Big[\eta_1\mathbf{J}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\mathbf{e}_0\left(t+\mathfrak{n}\right)+\eta_2\mathbf{J}^{\mathrm{T}}\mathbf{W}_{\mathrm{I}}\mathbf{e}_{\mathrm{I},0}\left(t+\mathfrak{n}\right)\Big].\end{aligned} \tag{36}$$

With use of $\mathbf{J}=\mathbf{C}\mathbf{A}^{\mathrm{n}-1}\mathbf{B}$ from (26), (36) can be rewritten as

$$\mathbf{l}_{a,1}\left(t+\mathfrak{n}\right)=\mathbf{l}_{a,0}\left(t+\mathfrak{n}\right)+\varepsilon\Big[\eta_1\mathbf{H}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\mathbf{e}_0\left(t+\mathfrak{n}\right)+\eta_2\mathbf{H}^{\mathrm{T}}\mathbf{W}_{\mathrm{I}}\mathbf{e}_{\mathrm{I},0}\left(t+\mathfrak{n}\right)\Big] \tag{37}$$

where $\mathbf{H}$ is the Hessian matrix defined as $\mathbf{H}=\mathbf{J}^{\mathrm{T}}\mathbf{J}$. It is very challenging to find the total error reduced by the PI- controller because the controller error is the function of output lengths. Thus, the gain parameters are considered individually to check the convergence condition of the controller. At first, the output actuator length due to proportional gain can be considered as

$$\mathbf{l}_{a,1}\left(t+\mathfrak{n}\right)=\mathbf{l}_{a,0}\left(t+\mathfrak{n}\right)+\varepsilon\eta_1\mathbf{H}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\mathbf{e}_0\left(t+\mathfrak{n}\right). \tag{38}$$

The proportional controller error can be calculated after the first iteration as

$$\mathbf{e}_1\left(t+\mathfrak{n}\right)=\mathbf{l}_d\left(t+\mathfrak{n}\right)-\mathbf{l}_{a,1}\left(t+\mathfrak{n}\right). \tag{39}$$

The proportional controller error in (36) can be rewritten by inserting (38) as

$$\begin{aligned}\mathbf{e}_1\left(t+\mathfrak{n}\right)&=\mathbf{l}_d\left(t+\mathfrak{n}\right)-\mathbf{l}_{a,0}\left(t+\mathfrak{n}\right)-\varepsilon\eta_1\mathbf{H}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\mathbf{e}_0\left(t+\mathfrak{n}\right)\\&=\mathbf{e}_0\left(t+\mathfrak{n}\right)-\varepsilon\eta_1\mathbf{H}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\mathbf{e}_0\left(t+\mathfrak{n}\right)\\&=\mathbf{e}_1\left(t+\mathfrak{n}\right)=\Big[\mathbf{I}-\varepsilon\eta_1\mathbf{H}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\Big]\mathbf{e}_0\left(t+\mathfrak{n}\right).\end{aligned} \tag{40}$$

Equation (40) indicates that the output length error after the first iteration will decrease if the condition $\left\|\mathbf{I}-\eta\varepsilon\mathbf{H}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\right\|<1$ is satisfied. Identically, a second iteration can be executed to reduce the output error even further. The error after the second iteration can be found as

$$\begin{aligned}\mathbf{e}_2\left(t+\mathfrak{n}\right)&=\Big[\mathbf{I}-\varepsilon\eta_1\mathbf{H}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\Big]\mathbf{e}_1\left(t+\mathfrak{n}\right)\\&=\Big[\mathbf{I}-\varepsilon\eta_1\mathbf{H}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\Big]^2\mathbf{e}_0\left(t+\mathfrak{n}\right).\end{aligned} \tag{41}$$

On continuation, the proportional gain error after an infinite number of iterations turns out to be

$$\mathbf{e}_\infty\left(t+\mathfrak{n}\right)=\Big[\mathbf{I}-\varepsilon\eta_1\mathbf{H}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\Big]^\infty\mathbf{e}_0\left(t+\mathfrak{n}\right). \tag{42}$$

From (42), the necessary and sufficient condition for the convergence of proportional gain values is given by

$$\left\|\mathbf{I}-\eta_1\varepsilon\mathbf{H}^{\mathrm{T}}\mathbf{W}_{\mathrm{P}}\right\|<1. \tag{43}$$

Similarly, the necessary and sufficient condition for the convergence of integral gain values is given by

$$\left\|\mathbf{I}-\eta_2\varepsilon\mathbf{H}^{\mathrm{T}}\mathbf{W}_{\mathrm{I}}\right\|<1. \tag{44}$$

Together, (43) and (44) ensure the convergence of the proposed control scheme.

V. EXPERIMENTAL TEST-BENCH AND PARAMETER IDENTIFICATION

The experimental setup, depicted in Fig. 10(a), consists of Robotino-XT and external OptiTrack motion sensor cameras. The live motion of the PBHA is captured by the OptiTrack system, which employs Prime 13 cameras, retro-reflective markers (See Fig. 10(b)), and Motive 2.0 software. The infrared light emitted by the cameras is reflected by the retro-reflective markers and recognized by the camera's sensor. The captured reflections are utilized to ascertain the 2D marker positions, which are utilized by Motive 2.0 software to estimate the corresponding 3D positions through reconstruction.

Motive 2.0 employs the direct linear transformation (DLT) method for 3D reconstruction. DLT is widely used in literature to find the 3D object space coordinates from the 2D camera plane. The reconstruction procedures are implemented based on two phases: i.e. a calibration phase and a reconstruction phase. For example, the mapping from a 3D object reference frame (x, y, z) to a 2D image reference frame (u, v) is given by [51]

$$u=\frac{a_0+a_1x+a_2y+a_3z}{1+a_4x+a_5y+a_6z},\qquad v=\frac{a_7+a_8x+a_9y+a_{10}z}{1+a_4x+a_5y+a_6z}, \tag{45}$$

where $a_0,\cdots,a_{10}$ are the physical parameters of the camera settings, which are fitted through least square regression during the calibration phase. Additional parameters appear if optical and de-centering distortion terms are considered [52]. Note that the DLT parameters are to be estimated through the calibration process. For calibration, a set of control points with known coordinates are employed. First of all, a calibration square (See. Fig. 11) with three markers were used to set the axes. Then calibration wand with markers at fixed distances, as shown in Fig. 11, is used. The calibration wand is randomly waved and spun, a process called wanding, within the control volume for calibration of the OptiTrack vision system through Motive 2.0, as shown in Fig. 10(c). The placement of the cameras as implemented here is the same as those used in [19]. Here, a total of four Prime 13 cameras and nine fixed markers (control points) were used together for reconstruction. The Prime 13 cameras are high-speed cameras with 1280 × 1024 spatial resolution, 240 fps frame rate, and 4.8 μm × 4.8 μm pixel size.

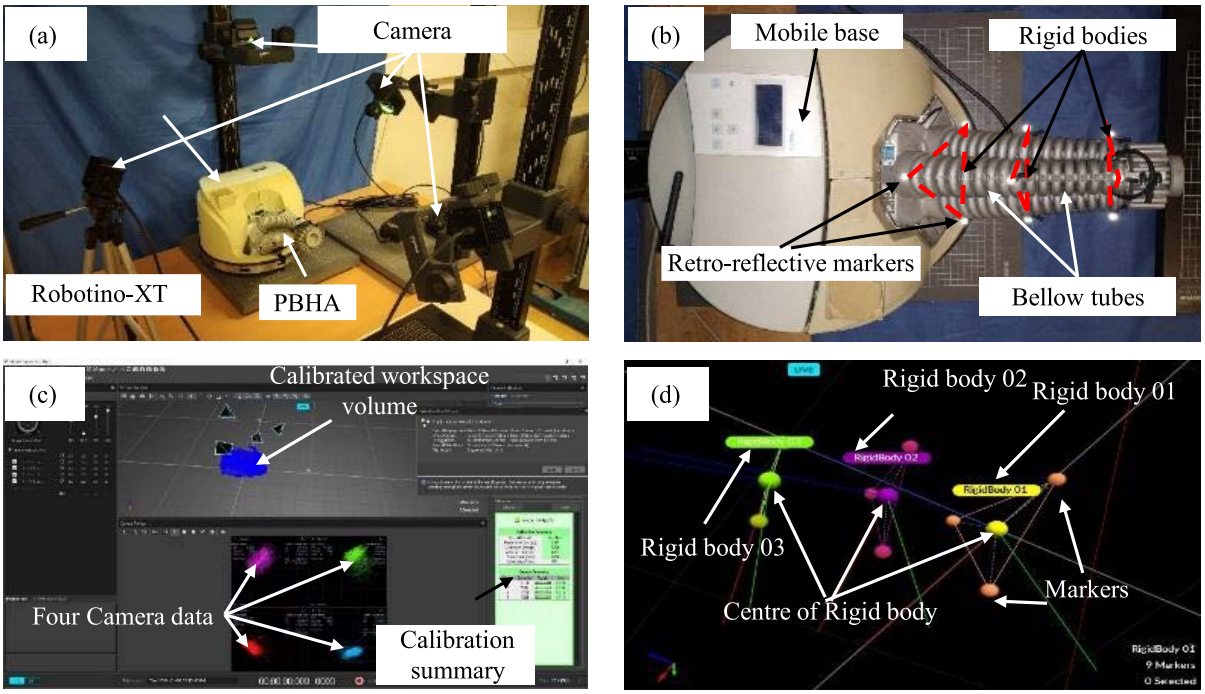

FIGURE 10. (a) Experimental test bench consisting of Robotino-XT with Prime 13 cameras, (b) The retro-reflective markers attached on the PBHA to form triangular shape rigid bodies, (c) OptiTrack calibrated workspace in Motive 2.0 interface with calibrated data summary, (d) Instantaneous position of the PBHA traced by OptiTrack system in Motive 2.0 interface.

FIGURE 11. CS-400 calibration square (left) and CW-500 calibration wand (right) from OptiTrack.

The Motive 2.0 software automatically extracts the 2D positions and executes 3D reconstruction. A single point (x, y, z) in 3D object reference plane is mapped to mapped to $\left(u^{(1)}, v^{(1)}\right)$, $\left(u^{(2)}, v^{(2)}\right)$, $\left(u^{(3)}, v^{(3)}\right)$ and $\left(u^{(4)}, v^{(4)}\right)$ points in the 2D image reference frame by four cameras. For the i-th camera, the DLT parameters are denoted $a_0^{(i)}$ to $a_{10}^{(i)}$. Then, using (45), the transformation from 2D frame to 3D frame for each marker seen by four cameras is written as [52]

$$\aleph \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{Bmatrix} a_0^{(1)} - u^{(1)} \\ a_7^{(1)} - v^{(1)} \\ \vdots \\ a_0^{(4)} - u^{(4)} \\ a_7^{(4)} - v^{(4)} \end{Bmatrix} \tag{46}$$

where

$$\aleph = \begin{bmatrix} u^{(1)}a_4^{(1)} - a_1^{(1)} & u^{(1)}a_5^{(1)} - a_2^{(1)} & u^{(1)}a_6^{(1)} - a_3^{(1)} \\ v^{(1)}a_4^{(1)} - a_8^{(1)} & v^{(1)}a_5^{(1)} - a_9^{(1)} & v^{(1)}a_6^{(1)} - a_{10}^{(1)} \\ \vdots & \vdots & \vdots \\ u^{(4)}a_4^{(4)} - a_1^{(4)} & u^{(4)}a_5^{(4)} - a_2^{(4)} & u^{(4)}a_6^{(4)} - a_3^{(4)} \\ v^{(4)}a_4^{(4)} - a_8^{(4)} & v^{(4)}a_5^{(4)} - a_9^{(4)} & v^{(4)}a_6^{(4)} - a_{10}^{(4)} \end{bmatrix}.$$

Note that if a marker is not seen by a camera, then the corresponding two rows of $\aleph$ are removed. At any point of time, each marker must be visible to at least three cameras. The above equations are solved with least square minimization, and the mean reconstruction error was found to be 0.137 mm from the calibration.

In this experiment, a total of nine markers were used: three markers on the base of the first segment, three at the base of the second segment, and three at the PBHA tip. Three markers were grouped together for every segment to define a rigid body (plane of a triangle), as shown in Fig. 10(b). Thus, a total of three rigid bodies (rigid body-01 at the base of the first segment, rigid body-02 at the base of the second segment, and rigid body-03 at the head of the second segment) as shown in Fig. 10(d) were used to track the motion of the PBHA backbone.

For the workspace of the tip of the PBHA (without the motion of base of the Robotino-XT), the input data is designed as per full factorial design with six factors and

four levels, which results in a total of 4^6 sets of inputs. The input factors indicate the pneumatic pressure in the six bellow tubes. The levels are chosen to be 0, 0.6, 1.2, and 1.8 bars, respectively. Based on each input data (six pressure values), the lengths for six actuator tubes were measured by the potentiometers, and the tip position and orientation were captured by the OptiTrack at a steady state. Here, the resulting workspace of the manipulator, as shown in Fig. 12, is determined by the OptiTrack vision system. The workspace is akin to the cloth volume in an open umbrella with a thicker canopy cloth and three prominent ribs.

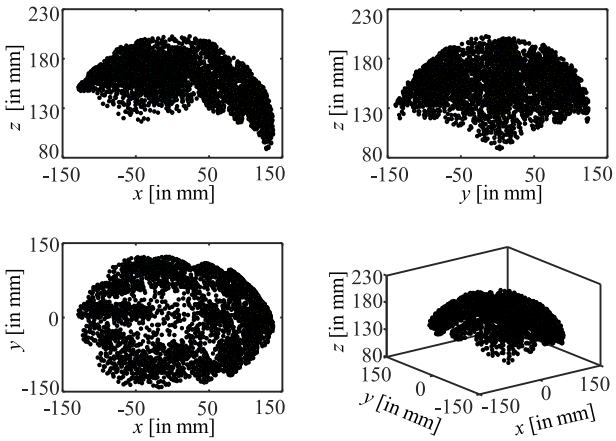

FIGURE 12. Workspace of the trunk of Robotino-XT.

The OptiTrack system reconstructed the coordinate points of each marker of a rigid body to calculate the position and orientation of the particular rigid body centroid. Note that each rigid body centroid lies on the backbone (a line connecting the centroid of the sections) of the trunk of the PBHA. The pose of the manipulator at any instant was determined by the OptiTrack system as $x^{(c)} = \left(x^{(1)} + x^{(2)} + x^{(3)}\right)/3$, $y^{(c)} = \left(y^{(1)} + y^{(2)} + y^{(3)}\right)/3$ and $z^{(c)} = \left(z^{(1)} + z^{(2)} + z^{(2)}\right)/3$, where superscripts (1) to (3) indicate the points on the rigid body arranged in an anti-clockwise sense as seen along the positive z-axis, and superscript (c) indicates the centroid or backbone. The orientation of the rigid body or the direction cosines of the unit normal to the plane of the rigid body is determined from vector algebra. Define vectors $\mathbf{V}_{12} = \left(x^{(2)} - x^{(1)}\right)\hat{i} + \left(y^{(2)} - y^{(1)}\right)\hat{j} + \left(z^{(2)} - z^{(1)}\right)\hat{k}$ and $\mathbf{V}_{13} = \left(x^{(3)} - x^{(1)}\right)\hat{i} + \left(y^{(3)} - y^{(1)}\right)\hat{j} + \left(z^{(3)} - z^{(1)}\right)\hat{k}$, and the corresponding unit vectors $\widehat{\mathbf{V}}_{12} = \mathbf{V}_{12}/|\mathbf{V}_{12}|$ and $\widehat{\mathbf{V}}_{13} = \mathbf{V}_{13}/|\mathbf{V}_{13}|$. Then $\widehat{\mathbf{V}}_{12} \times \widehat{\mathbf{V}}_{13} = \cos\theta_x\hat{i} + \cos\theta_y\hat{j} + \cos\theta_z\hat{k}$.

The inputs to the PBHA of the Robotino-XT are provided at intervals of at least two seconds. The pressure controller for the pneumatic proportional solenoid valves has low actuation gains to avoid overshoot and hence the response time to reach the steady-state is slow. The response is similar to that of a first-order system. The pressure controller includes cross-coupled gains to account for cross-coupling between the tubes. Though this reduces the speed of the actuator, it is indeed important because overshoot cannot be allowed in a safe human-robot interaction. The tip of the workspace is reached when all tubes have two bar pressure. The workspace is asymmetric due to the inclination of the PBHA (See Fig. 1) of Robotino-XT and the residual bending due to self-weight.

A. NEURAL NETWORK TUNING

For the IK of the PBHA, a suitable NN architecture has been chosen by trial and error, and it has been found that, with two hidden layers, the empirical risk function converges. The only restriction imposed on any NN model is that the output must be bounded (i.e., output $\in \mathcal{L}_\infty$). The activation functions for both the hidden layers are chosen to be sigmoid functions, whereas the linear activation function is chosen for the output layer. The number of hidden neurons (N_2) for both the hidden layers is kept the same and varies from 2 to 32 in steps 2. The training performance (MSE) of the NN by using the SL algorithm is compared with that by using the multilayer perceptron algorithm. The maximum time limit for any sub-network in the case of the SL algorithm is set to 130 seconds.

In literature, the ideal learning rate and the momentum coefficient are proposed in the range of (0, 1), and their optimum values should be selected through the trial and error method [53]. Thus, based on a grid search, the best learning rate and momentum coefficient for the networks are found to be 0.01 and 0.5, respectively. The best performance (training and validation) with training duration for all three networks are given in Table 1. The results show that the best training performance for NN_1 is obtained by considering $2\log_2(N_2)$ number of nodes in the hidden layer as suggested in [54],

TABLE 1. Performance of three layer NN trained by SL algorithm and multilayer perceptron.

		SL algorithm		Multilayer Perceptron	
NN type	**Number of nodes**	**MSE ($\times 10^{-5}$)**	**Time [s]**	**MSE ($\times 10^{-5}$)**	**Time [s]**
NN_1 $(\mathbf{x}_2)-(\mathbf{x}_1, \mathbf{x}_2')$	N_1: 3 N_3 : 6	Training: 0.21 Validation: 0.37 (N_2: 24)	81.29	Training: 4.98 Validation: ** (N_2: 18) [36]	84.9
NN_2 $(\mathbf{x}_1)-(\boldsymbol{\kappa}_1)$	N_1: 3 N_3 : 3	Training: 0.57 Validation: 0.71 (N_2: 20)	74.25	Training: 13.67 Validation: ** (N_2: 26)	137
NN_3 $(\mathbf{x}_2')-(\boldsymbol{\kappa}_2)$	N_1: 3 N_3 : 3	Training: 0.76 Validation: 1.03 (N_2: 20)	78.61	Training: 7.16 Validation: ** (N_2: 24)	127

and the best training performance for NN_2, NN_3 is obtained by considering $\left(2\log_2(N_2)-4\right)$ number of nodes in the hidden layer. Moreover, a comparison is also made with the multilayer perceptron method. It is evident from the results that the inverse solutions with SL algorithm-based NN model gives better result and is computationally more efficient for more extensive network sizes.

The computed pneumatic pressures from the NN model are used as input to the manipulator to check the accuracy of the system behaviour. The resulting actuator lengths at the steady-state position of the tip are computed, and the actual values are measured. The positional error at the manipulator tip is found to be nearly 7 mm in some instances, especially at the workspace boundary. This is an outcome of the non-linear hysteresis effect in the polyamide material used for the manipulator construction. Therefore, an open-loop IK controller would be erroneous, and a closed-loop controller is developed to eliminate the error. Moreover, the closed-loop controller is used here for the manipulator tip path tracking.

B. CONTROLLER PARAMETER TUNING

The gains of the feedback-feed forward PI controller (See Fig. 9) are adjusted carefully by selecting appropriate η, $\mathbf{W}_\mathrm{P}$, and $\mathbf{W}_\mathrm{I}$. On the basis of trial and error, the learning rate η is chosen as 0.01. It is noted that the first three diagonal elements of $\mathbf{W}_\mathrm{P}$ and $\mathbf{W}_\mathrm{I}$ correspond to the first segment of the manipulator, and the last three diagonal elements of $\mathbf{W}_\mathrm{P}$ and $\mathbf{W}_\mathrm{I}$ correspond to the second segment of the manipulator. Moreover, the controller continually changes the pressure set points when the command to move from one position to another is given. Therefore, the maximum response time of the controller to compensate for the errors was kept at three seconds during tuning of the controller gains so that the maximum percentage steady-state error for the step response is bounded within $\pm 2\%$ after three seconds. Note that the settling time of the base system (with open-loop control) was two seconds, and it was found that at least one second extra time is needed for error compensation.

VI. RESULTS AND DISCUSSION

In this section, the effectiveness of the developed kinematic control model for a class of BCM is verified on the PBHA of the Robotino-XT. For the purpose of validation, three different paths are considered. The first is a circular path, the second is a path like a double circle attached side by side with different radii, and the third is a simple 1:2 Lissajous curve, a part of which lies outside the workspace of PBHA with a stationary Robotino base.

A. CASE-1: CIRCULAR PATH

A circular end-effector path from the workspace of the PBHA is chosen to validate the developed control model. The chosen circle is of 115 mm radius, which is located at 165 mm from the base-plane along the z-axis of the PBHA. Discrete positions at 10 degrees intervals on the perimeter of the circle are chosen so that there are 37 positions through which the tip

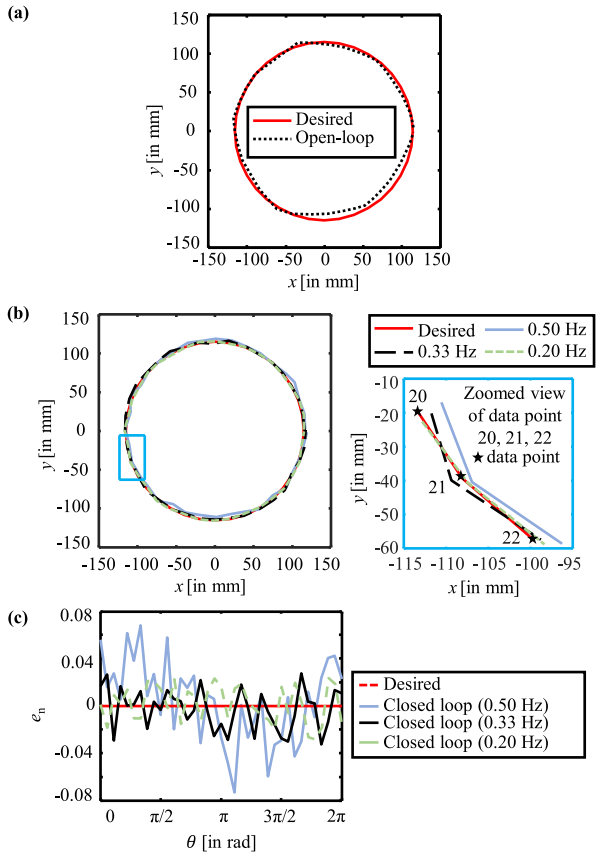

FIGURE 13. (a) Comparison of desired circular path and trajectory with open-loop control, (b) Comparison of desired circular path and trajectory with closed-loop control, and (c) Normalized error for radius.

moves to complete a full circle. The time interval to traverse between two positions is denoted by ΔT and the command update frequency is $\omega = 1/\Delta T$. The comparison between the tip position with open-loop control and the desired path is shown in Fig. 13(a). The same comparison along with a zoomed view of data points 20, 21, and 22 for the closed-loop control with three different values of update frequencies, is shown in Fig. 13(b).

The normalized error is defined as $e_n = (r_d - r_a)/r_d$ where r_d is the desired and r_a is the actual radius (distance from the center of the circle). This error is plotted in Fig. 13(c). The range of the normalized error e_n for 0.50 Hz frequency for closed-loop lies between -0.08 to 0.07, and the average e_n is found to be -0.0041. Similarly, e_n for 0.33 Hz frequency in closed-loop lies between ± 0.035, and the average e_n is found to be -0.0027. Moreover, by further reducing the frequency to. 0.20 Hz, e_n is found to be between ± 0.032, and the average e_n is found to be 0.0022. It is observed that the actual path with a lower command update frequency is closer to the desired path when compared with the higher frequency. The result for higher update frequency, e.g., 1 Hz, is at times worse than the open-loop control. This is because the closed-loop controller is not given enough time to

compensate for the positional errors. Moreover, by reducing the update frequency from 0.50 Hz to 0.33 Hz (approximately 34%), the normalized error reduced by 53%, whereas by reducing the update frequency from 0.33 Hz to 0.20 Hz (approximately 40%), the normalized error reduced by only 8%. Since the normalized error does not reduce much by reducing the update frequency, the lowest update frequency is chosen to be 0.33 Hz. Moreover, for a circle of 37 data points, 74 seconds of travel time can be saved by using 0.33 Hz frequency instead of 0.20 Hz.

The results for the circular trajectory obtained by using the present method for the same circular arc with 0.33 Hz update frequency are compared with the existing models in Table 2.

The tip position of the PBHA for both the open-loop and the closed-loop control is plotted with respect to the angular position in Fig 14. For the circular path, the x-coordinate of the tip follows a cosine curve and the y-coordinate of the tip follows a sine curve. A large error is clearly visible in Fig. 14 for some duration of the open-loop control. This error is due to the hysteresis effect, which occurs arbitrarily on repeated experiments, and the in-built open-loop controller cannot compensate for this error. However, the error is less for the three considered closed-loop update frequencies as compared to the open-loop control, and repeated experiments show similar results. Note that 0.2 Hz update frequency does not provide sufficient improvement over 0.33 Hz update frequency, and hence, results for below 0.33 Hz update frequency will not be discussed further.

TABLE 2. Recent contributions in continuum robot kinematics model.

Authors	Robot	Kinematics model	Accuracy
Escande *et al.* [17]	PBHA	Model-based: Constant curvature model	FK – 8 mm IK – NA
Mahl *et al.* [22]	BHA	Model-based: Variable curvature modelling	FK – 20 mm IK – 50 mm
Rolf *et al.* [35]	BHA	Model-free: Goal babbling learning	FK – 5 mm IK – 6 mm
Braganza *et al.* [28]	OctArm VI	Model-free: Feed-forward NN	FK – NA IK – 30 mm
Melingui *et al.* [34]	PBHA	Model-free: Adaptive NN	FK – 4 mm IK – 5 mm
Present work	PBHA	Hybrid Approach: Constant curvature and SL-NN	FK – 2.7 mm IK – 3.2 mm

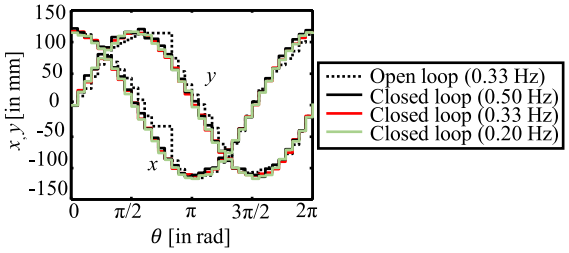

FIGURE 14. Tip position comparison of the PBHA for open-loop control and closed-loop control.

The PI-controller in this work is used to improve the dynamic performance, i.e., for faster rise time and smaller steady-state error. Instead of supplying the tip position information, the discrete PI-controller used in the closed-loop system is supplied with the difference/error in actuator length (difference of the desired length and actual length) of all the bellow tubes. Here, for each spatial coordinate of the robot tip, all desired actuator lengths of the PBHA are derived from the SL algorithm-NN model and manipulator sensors. The controller uses a Jacobian matrix to determine the change in the supplied pressures. The K_P and K_I gain values of the controller are adjusted at the start of each update step using (22) and (23), and those are kept constant within an update step, i.e., motion from one point to the other. The controller is provided with a sampling time of 0.05s, whereas the update time (reciprocal of update frequency) for trajectory points is much higher (2 seconds and 3 seconds, for 0.5 Hz and 0.33 Hz update frequencies, respectively). Thus, to complete the circular trajectory with 0.5 Hz and 0.33 Hz update frequencies takes 72 s and 108 s, respectively.

The transition from one point to the next two points on the tip trajectory is sown in Fig. 15, where the response of the system (actuator lengths) within the control period (from control point 20 to 21 and from control point 21 to 22) for update frequency of 0.33 Hz and 0.50 Hz are plotted together. Note that the sampling frequency is 0.05s. For each actuation command, there are 60 discrete control loop executions for 0.33 Hz update frequency and 40 discrete control loop executions for 0.50 Hz update frequency. Since the control step times are different (2 s and 3 s), time normalization is

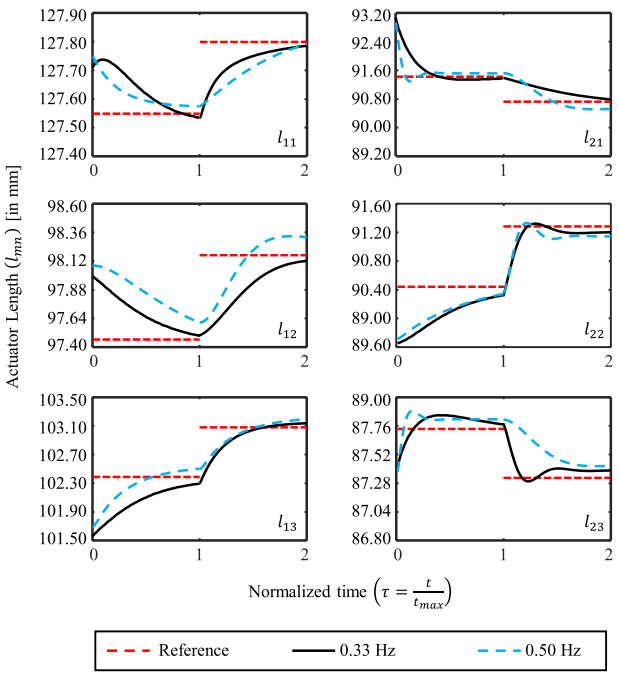

FIGURE 15. Actuator length variation of PBHA at $\tau = 0, 1, 2$, the PBHA tip is at data point 20, 21, and 22 respectively.

performed to plot the results in Fig. 15. The normalized time $\tau = t/t_{\max}$ where t is the actual time counted from point 20 (See Fig. 13(b)) and $t_{\max}$ is the step time. All six length variations of the PBHA from point 20 to point 22 (control points are selected from the circular trajectory in Fig. 13(b)) are plotted against normalized time, where $\tau = 0$, 1 and 2 at points 20, 21, and 22, respectively. Figure 13(c) shows that the residual errors are smaller in the case of 0.33 Hz update frequency as compared to a 0.50 Hz update frequency (note that the errors appear exaggerated due to the used scale). However, notice that in Fig. 15, the lengths are not converging to their desired values for most of the cases with 0.50 Hz update frequency. This error with a higher update frequency is because of the change in the input command to the PBHA before its actuator lengths can converge or settle to the desired lengths for the previous input command. Therefore, too fast actuation leads to more positional errors.

The residual error in actuator lengths depends on the nature and direction of the bending of the trunk. In different situations, some of the length parameters turn out to be well controlled, whereas the rest can have slightly more residual error. The transient overshoots are small, i.e., the tip position does not go too much outside the intended trajectory. This is an important control goal because otherwise, there can be serious implications during safety-critical applications like robotic surgery. To complete a task in less time, rather than using higher update rate, multiple robots sharing the task is preferred.

Apparently, the actuator lengths can converge closer to the desired set points if more time is given or the higher integral gain. However, more time reduces the update frequency and will make the robot motion slower. On the other hand, the proportional and integral gains are found from real-time controller tuning within each update step, and a better tuning is obtained when the learning rate parameter η is made even smaller. However, with such a change, the convergence of the controller tuning algorithm takes more time, and hence it is unsuitable for real-time application. This may be avoided with parallelization of the code, which is not attempted here. In the present work, due to limitations of the computational hardware and multi-threaded serial computing, the authors have stuck with a reasonably small (but not too small) learning rate and update frequency to maintain acceptable real-time performance while allowing for the minor residual errors in the actuator lengths. Further note that the controller is provided with no additional differential gain values because of the existing high damping resulting from the material properties of the PBHA (refer to the nearly over-damped nature of the responses in Fig. 15).

B. CASE-2: DOUBLE CIRCULAR PATH

This scenario considers a double circle-like path of the tip of the BCM, as shown in Fig. 16(a), where the tip trajectory moves in a clockwise direction for the bigger circle and in an anti-clockwise direction for the smaller circle. Here, the double circle consists of two circles joined together side by

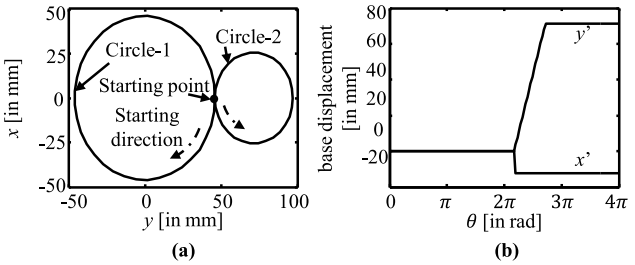

FIGURE 16. (a) Desired eight like path of tip of PBHA, (b) Path planning of Robotino-XT base for the desired path of double circle.

side, where the diameter of the first circle is 46 mm and that of the second circle is 25.5 mm. The coordinates of the first circle are chosen from the workspace of the PBHA, and the coordinates of the smaller circle are chosen outside the workspace of the PHBA. So, to track the desired path of the first circle, only actuation of the bellow tubes is required. However, to track the desired path of the second circle, simultaneous actuation of the bellow tubes and movement of the base is required.

The tip of the trunk is initially positioned at the intersection of the circles. It first completes the larger circle with clockwise rotation and then starts the smaller circle from the intersection point and completes it with anti-clockwise rotation. The starting point for the path is indicated in Fig. 16(a). The base motion of the Robotino-XT, as shown in Fig. 16(b), is planned such that the desired target point on the smaller circular path comes within the workspace of the trunk with minimum base movement. Here x', y' are the assigned coordinate system for the mobile base of Robotino-XT.

Figure 17(a) shows the path followed by the BCM tip for open-loop control with an update frequency of 0.33 Hz. It is quite natural that the nonlinearity and hysteresis of the

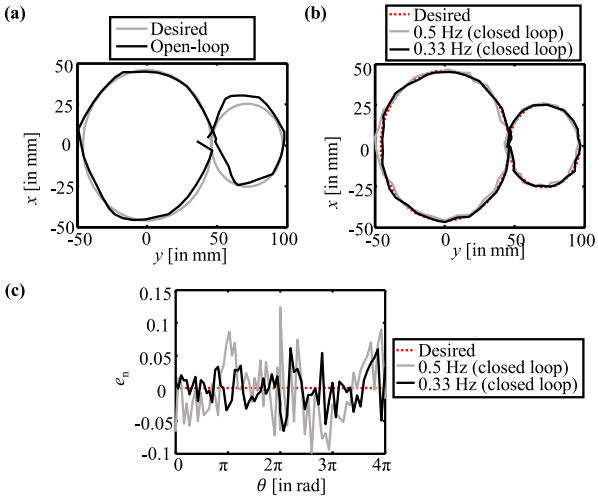

FIGURE 17. (a) Comparison of the desired path with the actual path (open-loop control), (b) Comparison of the desired path with the controlled path (closed-loop control), and (c) Closed-loop e_n, for the double circular path.

trunk affect the trajectory tracking accuracy. The closed-loop control is performed for two different update frequencies (0.33Hz and 0.5Hz), as shown in Fig. 17(b). The normalized error for the radius during closed-loop control is shown in Fig. 17(c) (the error plot for the open-loop control is omitted due to obvious presence of large error). It is observed that the normalized radius error in the case of the higher update frequency for the first circle ranges from -0.09 to 0.07, whereas the same for the second circle varies between ± 0.16. Similarly, in the case of lower update frequency for the first circle, e_n lies between -0.06 to 0.04, whereas the same for the second circle varies between ± 0.07. From several repeated experiments, it was found that a 33% decrease in update frequency (from 0.5 Hz to 0.33 Hz) reduced the average error by almost 70%. Further reduction in update frequency, say to 0.2 Hz, did not result in a significant reduction in the error.

C. CASE-3: LISSAJOUS PATTERN PATH

The parametric equations for the Lissajous curve, also termed as the Bowditch curve, are written as

$$\begin{aligned} x &= A_1 + A_2 \sin at \\ y &= B_1 + B_2 \sin (bt + \delta) \end{aligned} \tag{47}$$

where $(A_1,\ B_1)$ is the geometric center of the curve, (A_2 and B_2) are the x and y amplitudes, respectively, a and b are frequency constants, and δ is the phase shift in the y axis. Equation (47) describes a harmonic motion where $\left(a/b\right)$ plays an important role in defining the shape of the curve. For the present case, the simplest form of the Lissajous curve is taken into consideration, where $A_1 = 0$ mm, $B_1 = 100$ mm, $A_2 = 100$ mm, $B_2 = -200$ mm, $a = 2$ rad/s, $b = 1$ rad/s and $\delta = \pi/2$ rad. Inserting the parameters into (47) and varying t from 0 to 2π, the obtained Lissajous curve is shown in Fig. 18(a).

As per the path planning, the tip of the BCM starts its path from the starting point $(-100, 0)$ as indicated in Fig. 18(a), and rotates in the directions indicated therein. Including the start point and end points, a total 193 number of target points are considered on the discretized curve. However, many of the coordinates of the assumed Lissajous pattern do not lie within the workspace of the PHBA with a stationary base. So, the movement of the base is planned for the tip to reach the desired positions with the help of simultaneous actuation of the base and the bellows.

For planning the base movement, two imaginary circles of radius A_2, as shown in Fig. 18(a), are chosen. These circles lie within the workspace when the base frame origin is positioned at the center of the circle. Since both circles and the Lissajous curve have the same x axis amplitude of A_2, the robot base needs to be moved only to change the y position of the base frame to compensate for the gap between the circle and the Lissajous pattern. Note that the robot base can translate in the $x' - y'$ frame and the $y'-$axis is parallel to $y-$axis.

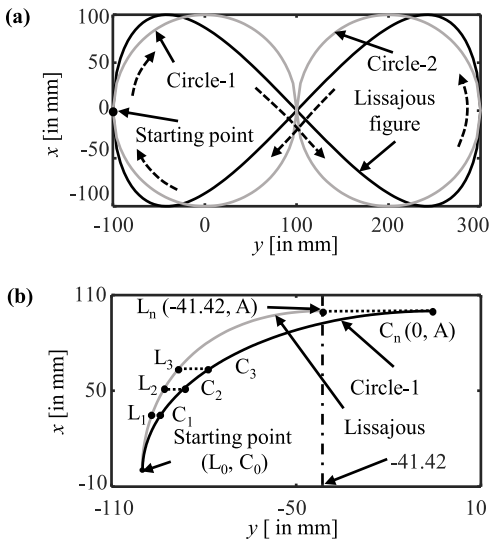

FIGURE 18. (a) The desired path for Lissajous pattern with a/b=2 and reference circle, (b) Zoomed view of Lissajous pattern and circle.

A zoomed section view of Fig. 18(a) is now shown in Fig. 18(b) so as to plan the necessary base movements. Here t is varied from 0 to $\pi/4$ for the Lissajous pattern and 0 to $\pi/4$ for the first circle. The plots for both the Lissajous pattern and circle start from $(-100, 0)$ and end at maximum x-coordinate, i.e. amplitude A. The maximum amplitude of the Lissajous pattern from Fig 18(b) is found by deriving a unified equation from (47) and applying the defined parameters as follows

$$x = 100 \sin \left\{ 2 \cos^{-1} \left(\frac{100 - y}{-200} \right) \right\}. \tag{48}$$

Differentiating (48) with respect to y, the maximum amplitude A_2 is found to be at $t = \pi/4$ and $y = -41.42$ whereas the first circle reaches its maximum amplitude A at $t = \pi/2$ and $y = 0$. The amplitude A is divided into n equal intervals, and horizontal lines are drawn, as shown in Fig 18(b), to design the base movement for the compensation of the gap between the two figures. Each horizontal line drawn has an intersection with both the Lissajous pattern and the circle. The intersections of those horizontal lines with the Lissajous pattern are denoted by points $L_0, L_1, L_2, \cdots, L_n$, and those with the circle are denoted by points $C_0, C_1, C_2, \cdots, C_n$. For each intersection of the line with the Lissajous pattern and the circle, the corresponding parametric variable t is determined, and the corresponding y-coordinates y_{L_i} and y_{C_i} $(i = 0, 1, 2, \cdots, n)$ values are also determined. The distance between the each y_{L_i} and y_{C_i} pair is calculated and is denoted by d_i. Note that $d_0 = L_0 - C_0 = 0$. Each subsequent step for the base motion will be compensated by $d_{i+1} - d_i$. Simultaneously, the bellow tubes of the trunk are actuated such that the tip of the trunk can reach C_i. In this way, the combined effect of both bellow actuation and base motion leads the trunk tip to reach the desired point L_i. Likewise,

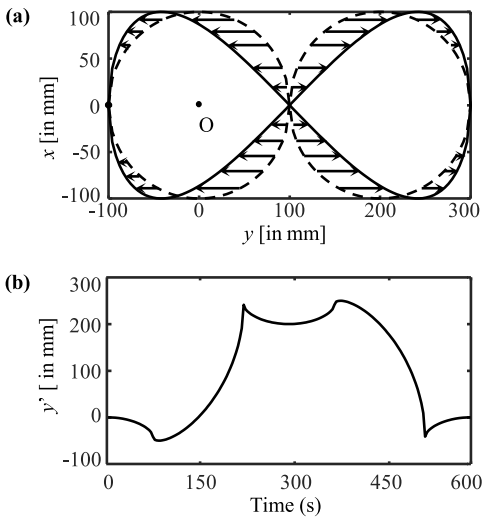

FIGURE 19. (a) Robotino-XT base movement direction from the origin, (b) Robotino-XT base movement.

the base motion compensation for the remaining points is planned as shown in Fig. 19.

The desired Lissajous path for the BCM is tracked with two different update frequencies (0.5Hz, 0.33Hz) for the feedback control. For each actuation process with a low update frequency (0.33Hz), the three seconds gap is provided. The tip of the manipulator reaches a steady state in these three seconds, and at this stage, the length data of all the bellows are acquired. The acquired data are further processed through a PI-controlled closed-loop to reach the desired path. The actual path traced by the tip of the BCM is plotted against the desired path for the Lissajous pattern in Fig. 20(a). Even with the requisite base motions, a good match is obtained

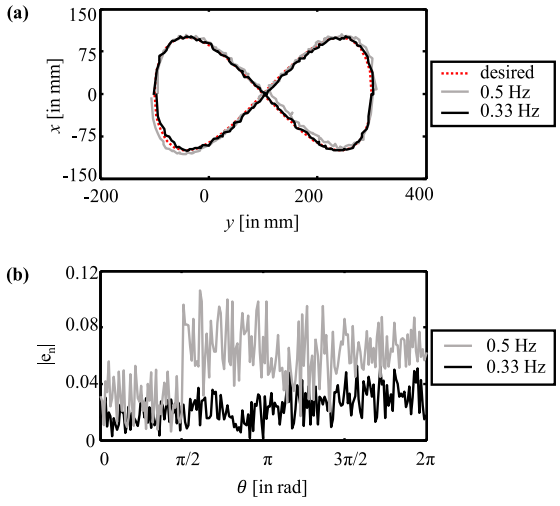

FIGURE 20. (a) Comparison of the desired path like a Lissajous pattern and actual path for both update frequencies, (b) absolute e_n of actual path for Lissajous pattern.

between the actual path and the desired path. The absolute normalized error $|e_n|$ between the actual and desired paths for the Lissajous figure, as shown in Fig. 20(b), is defined as

$$|e_n| = \left| \sqrt{\left(\frac{x_d - x_a}{A}\right)^2 - \left(\frac{y_d - y_a}{B}\right)^2} \right| \tag{49}$$

where, (x_d, y_d) is the desired coordinate for the tip and (x_a, y_a) is the actual coordinate of the tip. The z-direction positional error is not considered in Eq. (49) because it is too small (practically absent). For high update frequency, Fig. 20(b) shows that $|e_n| < 0.1$ for the entire path and the mean is about 0.055. For low update frequency, $|e_n| < 0.05$ with a mean of 0.025. The results show that the error is more when the base acceleration is more (See Fig. 19(b) for base displacement profile). This is because the base motions apply inertia forces on the trunk, which is an un-modeled dynamics so far as this article is concerned.

VII. CONCLUSION

In comparison to existing literature, a more accurate method to navigate the tip of a BCM in real-time is proposed. Instead of using complex nonlinear equations, the IK model is solved by combining model-free and a simple model-based technique. The SL algorithm-based NN is used to solve the inverse problem of the BCMs. The effectiveness of the proposed algorithm's accuracy, time duration, and stability compared to traditional multilayer perceptron is discussed in detail. Since the material hysteresis has a predominant effect on the BCM and its response cannot be cyclically repeated, a new real-time closed-loop control scheme by using a Jacobian-based PI-controller in cascaded form is proposed. The convergence criteria, along with the pneumatic pressure saturation of the designed controller, are also provided. The proposed model is validated with an elephant trunk-like BCM called Robotino-XT. Three different paths (a circle, a double circle, and a Lissajous pattern) are taken into consideration to show the accuracy of the model. The closed-loop control model gives a much better result than the open-loop controller. A comparison study is also provided for different update frequencies, and it is observed that the accuracy is much better for a lower update frequency. Moreover, with the proposed model and lower update frequency, for all the cases taken into account, the maximum normalized error that occurs is within $\pm 1.5\%$ of the manipulator length, which is reasonable for a flexible manipulator under inertial load. When some part of the path falls outside the workspace of the manipulator (e.g., the double circle and Lissajous patterns), the path of the mobile base is planned accordingly. The control of the mobile base is performed by a low-level controller inbuilt with the Robotino-XT.

The present model is based on kinematic considerations. Therefore, it is observed that with smaller update time intervals (i.e., larger update frequency) between two target points in the trajectory, the positional accuracy reduces. Two reasons for this inaccuracy are that the low-level controller has

a significant response time (due to pneumatic actuation), and the inertia forces are not considered in the model. It is expected that an inverse dynamics model would represent the effects of the response time of the pneumatic actuation and the inertia forces. So far, no such attempt has been made for the kind of soft robot considered in this study, and it can be a promising area for future research, especially to improve positional accuracy with high-speed trajectory tracking.

The current study only focuses on the path control without any imposed optimality condition, such as least energy consumption. The primary aim of this research is to highlight the use of the hybrid modeling approach and the cascaded control where one controller specifies the set-point for another. A more complex controller may perform better in terms of positional accuracy, provided that the associated computations can be performed in real-time without delaying the robot's response. Therefore, in the future, optimality conditions may be used for path planning while keeping real-time performance in view. As further future work, tip force control with respect to the loading conditions and interaction between two or more manipulators will be addressed.

ACKNOWLEDGMENT

The authors would like to thank Inderjeet Singh, Lakhal Othman, Steeve Mbakop, and Rochdi Merzouki for their help in conducting the experiments at Université de Lille.

SUPPLEMENTARY MATERIAL

Supplementary Video: Real-time experiments on Robotino-XT.

REFERENCES

[1] J. Mendoza-Mendoza, V. J. Gonzalez-Villela, C. Aguilar-Ibanez, S. Suarez-Castanon, and L. Fonseca-Ruiz, "Snake aerial manipulators: A review," *IEEE Access*, vol. 8, pp. 28222–28241, 2020, doi: 10.1109/ACCESS.2020.2971247.

[2] M. Wehner, R. L. Truby, D. J. Fitzgerald, B. Mosadegh, G. M. Whitesides, J. A. Lewis, and R. J. Wood, "An integrated design and fabrication strategy for entirely soft, autonomous robots," *Nature*, vol. 536, no. 7617, pp. 451–455, Aug. 2016, doi: 10.1038/nature19100.

[3] M. K. Mishra, S. Ghosal, A. K. Samantaray, and G. Chakraborty, "Jacobian-based inverse kinematics analysis of a pneumatic actuated continuum manipulator," in *Advances in Industrial Machines and Mechanisms*. Singapore: Springer, Jul. 2021, pp. 3–13, doi: 10.1007/978-981-16-1769-0_1.

[4] P. Liu, M. N. Huda, L. Sun, and H. Yu, "A survey on underactuated robotic systems: Bio-inspiration, trajectory planning and control," *Mechatronics*, vol. 72, Dec. 2020, Art. no. 102443, doi: 10.1016/j.mechatronics.2020.102443.

[5] H. Wang, X. Wang, W. Yang, Z. Du, and Z. Yan, "Construction of controller model of notch continuum manipulator for laryngeal surgery based on hybrid method," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 2, pp. 1022–1032, Apr. 2021, doi: 10.1109/TMECH.2020.3015133.

[6] Z. Samadikhoshkho, S. Ghorbani, and F. Janabi-Sharifi, "Modeling and control of aerial continuum manipulation systems: A flying continuum robot paradigm," *IEEE Access*, vol. 8, pp. 176883–176894. Singapore: Springer, 2020, doi: 10.1109/ACCESS.2020.3026279.

[7] M. K. Mishra, A. K. Samantaray, and G. Chakraborty, "Fractional-order Bouc-Wen hysteresis model for pneumatically actuated continuum manipulator," *Mechanism Mach. Theory*, vol. 173, pp. 1–21, Mar. 2022, doi: 10.1016/j.mechmachtheory.2022.104841.

[8] I. A. Seleem, H. El-Hussieny, S. F. M. Assal, and H. Ishii, "Development and stability analysis of an imitation learning-based pose planning approach for multi-section continuum robot," *IEEE Access*, vol. 8, pp. 99366–99379, 2020, doi: 10.1109/ACCESS.2020.2997636.

[9] V. Pachouri, P. M. Pathak, M. K. Mishra, A. K. Samantharay, R. Merzouki, and B. O. Bouamama, "Workspace evaluation of Robotino-XT under reconfiguration," in *Machines, Mechanism and Robotics*. Singapore: Springer, Jul. 2021, doi: 10.1007/978-981-16-0550-5_25.

[10] K. Choi, J. Kwon, T. Lee, C. Park, J. Pyo, C. Lee, S. Lee, I. Kim, S. Seok, Y. J. Kim, and F. C. Park, "A hybrid dynamic model for the AMBIDEX tendon-driven manipulator," *Mechatronics*, vol. 69, Aug. 2020, Art. no. 102398, doi: 10.1016/j.mechatronics.2020.102398.

[11] M. Goharimanesh, A. Mehrkish, and F. Janabi-Sharifi, "A fuzzy reinforcement learning approach for continuum robot control," *J. Intell. Robot. Syst.*, vol. 100, nos. 3–4, pp. 809–826, Aug. 2020, doi: 10.1007/s10846-020-01237-6.

[12] A. Melingui, J. J.-B. Mvogo Ahanda, O. Lakhal, J. B. Mbede, and R. Merzouki, "Adaptive algorithms for performance improvement of a class of continuum manipulators," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 9, pp. 1531–1541, Sep. 2018, doi: 10.1109/TSMC.2017.2678605.

[13] J. L. C. Santiago, I. S. Godage, P. Gonthina, and I. D. Walker, "Soft robots and kangaroo tails: Modulating compliance in continuum structures through mechanical layer jamming," *Soft Robot.*, vol. 3, no. 2, pp. 54–63, Jun. 2016, doi: 10.1089/soro.2015.0021.

[14] M. K. Mishra, A. K. Samantaray, G. Chakraborty, A. Jain, P. M. Pathak, and R. Merzouki, "Dynamic modelling of an elephant trunk like flexible bionic manipulator," in *Proc. ASME Int. Mech. Eng. Congr. Expo.*, vol. 4. Salt Lake City, UT, USA, Jan. 2020, pp. 1–9, doi: 10.1115/IMECE2019-11113.

[15] B. A. Jones, W. McMahan, and I. Walker, "Design and analysis of a novel pneumatic manipulator," *IFAC Proc. Volumes*, vol. 37, no. 14, pp. 687–692, Sep. 2004, doi: 10.1016/S1474-6670(17)31183-7.

[16] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1661–1683, Jun. 2010, doi: 10.1177/0278364910368147.

[17] C. Escande, T. Chettibi, R. Merzouki, V. Coelen, and P. M. Pathak, "Kinematic calibration of a multisection bionic manipulator," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 663–674, Apr. 2014, doi: 10.1109/TMECH.2014.2313741.

[18] J. Sun and J. Zhao, "Physics-based modeling of twisted-and-coiled actuators using Cosserat rod theory," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 779–796, Apr. 2021, doi: 10.1109/TRO.2021.3104238.

[19] F. Janabi-Sharifi, A. Jalali, and I. D. Walker, "Cosserat rod-based dynamic modeling of tendon-driven continuum robots: A tutorial," *IEEE Access*, vol. 9, pp. 68703–68719, 2021, doi: 10.1109/ACCESS.2021.3077186.

[20] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *J. Robot. Syst.*, vol. 20, no. 2, pp. 45–63, Feb. 2003, doi: 10.1002/rob.10070.

[21] X. Huang, J. Zou, and G. Gu, "Kinematic modeling and control of variable curvature soft continuum robots," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 6, pp. 3175–3185, Dec. 2021, doi: 10.1109/TMECH.2021.3055339.

[22] T. Mahl, A. Hildebrandt, and O. Sawodny, "A variable curvature continuum kinematics for kinematic control of the bionic handling assistant," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 935–949, Aug. 2014, doi: 10.1109/TRO.2014.2314777.

[23] I. Singh, Y. Amara, A. Melingui, P. M. Pathak, and R. Merzouki, "Modeling of continuum manipulators using Pythagorean hodograph curves," *Soft Robot.*, vol. 5, no. 4, pp. 425–442, Aug. 2018, doi: 10.1089/soro.2017.0111.

[24] H. Wang, Z. Du, W. Yang, Z. Y. Yan, and X. Wang, "Variable stiffness model construction and simulation verification of coupled notch continuum manipulator," *IEEE Access*, vol. 7, pp. 154761–154769, 2019, doi: 10.1109/ACCESS.2019.2949376.

[25] M. K. Mishra, A. K. Samantaray, G. Chakraborty, V. Pachouri, P. M. Pathak, and R. Merzouki, "Kinematics model of bionic manipulator by using elliptic integral approach," in *Machines, Mechanism and Robotics*, Jul. 2021, pp. 1–7, doi: 10.1007/978-981-16-0550-5_30.

[26] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, and C. Laschi, "Neural network and Jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature," *IEEE Trans. Robot.*, vol. 31, no. 4, pp. 823–834, Apr. 2015, doi: 10.1109/TRO.2015.2428511.

[27] H. Abidi, G. Gerboni, M. Brancadoro, J. Fras, A. Diodato, M. Cianchetti, H. Wurdemann, K. Althoefer, and A. Menciassi, "Highly dexterous 2-module soft robot for intra-organ navigation in minimally invasive surgery," *Int. J. Med. Robot. Comput. Assist. Surg.*, vol. 14, no. 1, pp. 1–9, Dec. 2017, doi: 10.1002/rcs.1875.

[28] D. Braganza, D. M. Dawson, I. D. Walker, and N. Nath, "A neural network controller for continuum robots," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1270–1277, Dec. 2007, doi: 10.1109/CDC.2006.377452.
[29] S. Xu, B. He, Y. Zhou, Z. Wang, and C. Zhang, "A hybrid position/force control method for a continuum robot with robotic and environmental compliance," *IEEE Access*, vol. 7, pp. 100467–100479, 2019, doi: 10.1109/ACCESS.2019.2928572.
[30] I. S. Godage, E. Guglielmino, D. T. Branson, G. A. Medrano-Cerda, and D. G. Caldwell, "Novel modal approach for kinematics of multisection continuum arms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, Sep. 2011, pp. 1093–1098, doi: 10.1109/IROS.2011.6094477.
[31] K. Xu and N. Simaan, "Analytic formulation for kinematics, statics, and shape restoration of multibackbone continuum robots via elliptic integrals," *J. Mech. Robot.*, vol. 2, no. 1, pp. 1–13, Feb. 2010, doi: 10.1115/1.4000519.
[32] S. Neppalli, M. A. Csencsits, B. A. Jones, and I. D. Walker, "Closed-form inverse kinematics for continuum manipulators," *Adv. Robot.*, vol. 23, no. 15, pp. 2077–2091, Apr. 2012, doi: 10.1163/016918609X12529299964101.
[33] A. Melingui, R. Merzouki, J. B. Mbede, C. Escande, B. Daachi, and N. Benoudjit, "Qualitative approach for inverse kinematic modeling of a compact bionic handling assistant trunk," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Beijing, China, Jul. 2014, pp. 754–761, doi: 10.1109/IJCNN.2014.6889947.
[34] A. Melingui, O. Lakhal, B. Daachi, J. B. Mbede, and R. Merzouki, "Adaptive neural network control of a compact bionic handling arm," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 6, pp. 2862–2875, Dec. 2015, doi: 10.1109/TMECH.2015.2396114.
[35] M. Rolf, J. J. Steil, and M. Gienger, "Goal babbling permits direct learning of inverse kinematics," *IEEE Trans. Auto. Mental Develop.*, vol. 2, no. 3, pp. 216–229, Sep. 2010, doi: 10.1109/TAMD.2010.2062511.
[36] O. Lakhal, A. Melingui, and R. Merzouki, "Hybrid approach for modeling and solving of kinematics of a compact bionic handling assistant manipulator," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 3, pp. 1326–1335, Jun. 2016, doi: 10.1109/TMECH.2015.2490180.
[37] Q. Zhao, J. Lai, K. Huang, X. Hu, and H. K. Chu, "Shape estimation and control of a soft continuum robot under external payloads," *IEEE/ASME Trans. Mechatronics*, early access, Oct. 19, 2021, doi: 10.1109/TMECH.2021.3116970.
[38] D. B. Camarillo, C. R. Carlson, and J. K. Salisbury, "Configuration tracking for continuum manipulators with coupled tendon drive," *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 798–808, Aug. 2009, doi: 10.1109/TRO.2009.2022426.
[39] R. S. Penning, J. Jung, N. J. Ferrier, and M. R. Zinn, "An evaluation of closed-loop control options for continuum manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 5392–5397, doi: 10.1109/ICRA.2012.6224735.
[40] C. Yang, S. Geng, I. Walker, D. T. Branson, J. Liu, J. S. Dai, and R. Kang, "Geometric constraint-based modeling and analysis of a novel continuum robot with shape memory alloy initiated variable stiffness," *Int. J. Robot. Res.*, vol. 39, no. 14, pp. 1620–1634, Apr. 2020, doi: 10.1177/0278364920913929.
[41] M. Dehghani and S. A. A. Moosavian, "Compact modeling of spatial continuum robotic arms towards real-time control," *Adv. Robot.*, vol. 28, no. 1, pp. 15–26, Nov. 2013, doi: 10.1080/01691864.2013.854452.
[42] C. Lopez-Franco, J. Hernandez-Barragan, A. Y. Alanis, and N. Arana-Daniel, "A soft computing approach for inverse kinematics of robot manipulators," *Eng. Appl. Artif. Intell.*, vol. 74, pp. 104–120, Sep. 2018, doi: 10.1016/j.engappai.2018.06.001.
[43] F. L. Lewis, S. Jagannathan, and A. Yesildirak, *Neural Network Control of Robot Manipulators and Non-Linear Systems*. Philadelphia, PA, USA: Taylor & Francis, 1999, pp. 223–278.
[44] A. Liu, Y. Zhang, H. Zhao, S. Wang, and D. Sun, "Neural network control system of cooperative robot based on genetic algorithms," *Neural Comput. Appl.*, vol. 33, pp. 8217–8226, May 2020, doi: 10.1007/s00521-020-04952-1.
[45] M. Ivanescu, N. Popescu, and D. Popescu, "A variable length tentacle manipulator control system," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 2005, pp. 3274–3279, doi: 10.1109/ROBOT.2005.1570615.
[46] M. C. Yip and D. B. Camarillo, "Model-less feedback control of continuum manipulators in constrained environments," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 880–889, Aug. 2014, doi: 10.1109/TRO.2014.2309194.
[47] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 124–134, Feb. 2019, doi: 10.1109/TRO.2018.2878318.
[48] B. Choi, J.-H. Lee, and D.-H. Kim, "Solving local minima problem with large number of hidden nodes on two-layered feed-forward artificial neural networks," *Neurocomputing*, vol. 71, nos. 16–18, pp. 3640–3643, 2008, doi: 10.1016/j.neucom.2008.04.004.
[49] W. Wu, H. Shao, and Z. Li, "Convergence of batch BP algorithm with penalty for FNN training," in *Neural Information Processing*. New York, NY, USA: Springer, 2006, pp. 562–569.
[50] W. Shen, G. Yang, T. Zheng, Y. Wang, K. Yang, and Z. Fang, "An accuracy enhancement method for a cable-driven continuum robot with a flexible backbone," *IEEE Access*, vol. 8, pp. 37474–37481, 2020, doi: 10.1109/ACCESS.2020.2975087.
[51] L. Chen, C. W. Armstrong, and D. D. Raftopoulos, "An investigation on the accuracy of three-dimensional space reconstruction using the direct linear transformation technique," *J. Biomech.*, vol. 27, no. 4, pp. 493–500, 1994, doi: 10.1016/0021-9290(94)90024-8.
[52] Y-H. Kwon. *DLT Method*. Accessed: Feb. 3, 2022. [Online]. Available: http://www.kwon3d.com/theory/dlt/dlt.html
[53] H. R. Maier and G. C. Dandy, "The effect of internal parameters and geometry on the performance of back-propagation neural networks: An empirical study," *Environ. Model.*, vol. 13, no. 2, pp. 193–209, May 1998, doi: 10.1016/S1364-8152(98)00020-6.
[54] H.-F. Yang, T. S. Dillon, and Y.-P. P. Chen, "Optimized structure of the traffic flow forecasting model with a deep learning approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2371–2381, Oct. 2017, doi: 10.1109/TNNLS.2016.2574840.

MRUNAL KANTI MISHRA received the B.Tech. degree in mechanical engineering from Siksha 'O' Anusandhan University, Bhubaneswar, India, in 2013, and the M.Tech. degree (Hons.) in mechanical engineering from the Veer Surendra Sai University of Technology, Burla, India, in 2016. He is currently pursuing the Ph.D. degree with the Systems, Dynamics and Control Laboratory, Department of Mechanical Engineering, Indian Institute of Technology (IIT) Kharagpur. He served as a Lecturer with the Veer Surendra Sai University of Technology, in 2016. His research interests include robot kinematics and dynamics, artificial intelligence, and parametric instability.

ARUN KUMAR SAMANTARAY received the B.Tech. degree from the College of Engineering and Technology (CET), Bhubaneswar, in 1989, and the M.Tech. degree in machine dynamics and the Ph.D. degree in mechanical engineering from the Indian Institute of Technology (IIT) Kharagpur, in 1991 and 1996, respectively. He worked as a Project Manager with High-Tech Consultants, from 1996 to 2000, and as a CNRS Researcher, France, from 2001 to 2004. He joined the Department of Mechanical Engineering, IIT Kharagpur, as a Faculty Member, in 2004, where he is currently a Full Professor and in-charge of the Systems, Dynamics and Control Laboratory. His research interests include systems and control, fault diagnosis, automation, non-linear mechanics, robotics, rotor dynamics, and vehicle dynamics.

GOUTAM CHAKRABORTY received the B.Tech. degree in mechanical engineering and the M.Tech. degree in machine dynamics from the Indian Institute of Technology (IIT) Kharagpur, in 1992 and 1994, respectively, and the Ph.D. degree from IIT Kanpur, in 1999. Thereafter, he did a postdoctoral research at the University of Maryland and TU Darmstadt. He joined the Department of Mechanical Engineering, IIT Kharagpur, as a Faculty Member, in 2003, where he is currently a Full Professor. His research interests include dynamics and control, and vibration analysis.