SOFT ROBOTICS
Volume 4, Number 3, 2017
© Mary Ann Liebert, Inc.
DOI: 10.1089/soro.2016.0051

ORIGINAL ARTICLE

Learning Closed Loop Kinematic Controllers for Continuum Manipulators in Unstructured Environments

Thomas George Thuruthel, Egidio Falotico, Mariangela Manti, Andrea Pratesi, Matteo Cianchetti, and Cecilia Laschi

Abstract

This article introduces a machine-learning-based approach for closed loop kinematic control of continuum manipulators in the task space. For this purpose, we propose a unique formulation for learning the inverse kinematics of a continuum manipulator while integrating end-effector feedback. We demonstrate that this model-free approach for kinematic control is very well suited for nonlinear stochastic continuum robots. The article addresses problems that are vital for practical realization of machine-learning techniques. The primary objective is to solve the redundancy problem while making the algorithm scalable, fast, and tolerant to stochasticity, requiring minimal sensor elements and involving few open parameters for tuning. In addition, we demonstrate that the proposed controller can exhibit adaptive behavior in the presence of external forces and in an unstructured environment with the help of the morphological properties of the manipulator. Experimental validation of the proposed controller is done on a six-degree-of-freedom tendon-driven manipulator for pose control of the end effector in three-dimensional space with and without external forces. The experimental results exhibit accurate, reliable, and adaptive behavior of the proposed system, which appears suitable for the field of continuum service robots.

Keywords: artificial neural networks, continuum robot, kinematic control, machine learning, morphological computation, unstructured environment

Introduction

The growth of soft robotics has been driven by new scientific paradigms and applications, which offer a radical technological solution for well-known robotics problems. It has led to an alternate view on intelligence, incorporating the role of the physical body for fast, efficient, and robust behavior.[1,2] Recent years have seen the increasing trend of employing soft robots for grasping tasks,[3,4] adaptive locomotion in land[5–7] and water,[8,9] flexible prosthesis,[10] and combined tasks.[11] Most of the research in the field of soft robotics is currently being done on their design, material, and fabrication. However, they are still lacking practical controllers that are employable in the real world.

Many of the traditional approaches for design, modeling, and control, which have been so far effective for rigid robots, have become ineffective or incompetent for soft robots. The most commonly used kinematic modeling approach for soft robots is based on the constant curvature approximation.[12] More rigorous geometrically exact models were proposed by Renda *et al.*[13,14] However, unlike rigid robots, the kinematic models of soft robots are also dependent on environmental variables because of their numerous underactuated and compliant degrees of freedom (DoFs). The existing controllers based on forward models are either too simple to account for external constraints or become too complex with the addition of external constraints.

There have been surprisingly few works on task space control of continuum manipulators in an unstructured environment, even though they are potentially better suited than rigid robots in such conditions.[15] One of the first task space controllers with sensory feedback was proposed by Camarillo *et al.*[16] using analytical models to estimate the forward kinematics. Similarly, Kapadia *et al.* later introduced a model-based task space controller while assuming that the kinematic and dynamic model can be obtained and easily observable.[17,18]

The BioRobotics Institute, Scuola Superiore Sant'Anna, Pisa, Italy.

Joint space controllers based on the constant curvature approximation were proposed by Penning *et al.*[19,20] However, all these approaches still relied on analytical models that are heavily prone to modeling error and are invalid in the presence of environmental constraints and, hence, not suitable for more practical applications. Furthermore, system identification for the developed models is cumbersome and susceptible to errors.

Correspondingly, Marchese *et al.* developed closed loop control algorithms for fully soft manipulators based on piecewise constant curvature approximation[21] and extended them for whole arm motion planning in a known environment.[22] Interestingly, the motion planning was done while considering the manipulator to adapt to the environment. Again, such methods are heavily dependent on the model formulation and sensing capabilities and are based only on the closed loop control of the segment curvature and not the end-effector position. Alternatively, real-time control of soft robots was successfully implemented by using a finite element-based method (FEM) by Duriez.[23] Although they promise to be a very accurate representation of the actual model and can possibly incorporate known environmental constraints, their scalability to higher-order systems is questionable. Furthermore, FEM-based approaches become much more computationally expensive if they are to model external contacts too. Few researchers have also focused on dynamic control of continuum manipulators, however, still relying on the constant curvature approximation.[24]

On the other hand, the underlying complexity and variability of continuum robots has prompted researchers to investigate the viability of model-free methodologies for control. Recognizing the fact that accurate analytical models are difficult to develop for any practical application, it is reasonable to lean toward model-free methods for modeling and control. A ''model-less'' controller for continuum robot was formulated by Yip and Camarillo by using empirical estimates of the Jacobian matrix for task space control.[25,26] Although these type of controllers are effective in unstructured environments, they require online estimation of the Jacobian matrix and are, hence, slow.

Model-free controllers based on machine learning are a promising alternative considering their potential to adapt to altering conditions and generalize well between observed data, even in the presence of noise. Not only are they applicable to a much broader variety of robots, but they can also be faster to deploy and provide the user the freedom to determine the underlying complexity of the model implicitly.

The earliest usage of machine-learning techniques in the field of continuum robots was for the compensation of unknown dynamics of systems.[27] Later, Giorelli *et al.*[28] proposed the use of a feed-forward neural network for learning the inverse statics of a soft cable-driven three DoF manipulator. Their study indicated the effectiveness of machine-learning-based controllers over even a thoughtfully constructed analytical model for continuum robots. Even so, their approach cannot be applied to redundant or high-dimensional continuum manipulators. Rolf *et al.*[29] presented a novel approach called goal babbling for bootstrapping inverse models in high-dimensional systems. A similar approach using reinforcement learning was proposed by Ansari *et al.*[30] Such approaches scale well for high-dimensional systems, although they are not the most efficient. Again, learning one solution at a time restricts the full capability of a continuum robot and it becomes invalid in the presence of external factors. Another interesting approach for learning the inverse kinematics (IK) was proposed by Melingui *et al.*[31] Their method involves learning the forward model and inverting it by using distal supervised learning. In addition, they have an adaptive subcontroller for compensating nonstationary kinetics. Again, these methods learn a particular solution to the IK problem and are, therefore, invalid in the presence of obstacles; moreover, they do not scale well for higher-dimensional systems.

Understandably, machine-learning-based approaches for obtaining inverse models are more developed for the conventional rigid robots. Thus, a small digression is made to comment on the most prevalent machine-learning techniques developed for rigid robots. A common problem associated with learning the IK of a redundant robot is the non-uniqueness of the solution. Coupled with the fact that the set of all the possible solutions form a concave set makes direct learning of the IK mapping hard.[32] Therefore, it is necessary to have additional constraints/strategies to restrict/select the appropriate solutions.

One of the earliest approaches for obtaining valid IK solutions by machine learning was proposed with the distal supervised learning method.[33] A more recent and popular approach is to learn the IK problem in the velocity level by using differential IK.[32,34] This is driven by the insight that the solution set of the differential IK problem forms a convex set locally. However, working on a velocity level would require integration over a predefined path to get solutions to positional targets. Consequently, few researchers have tried to develop direct learning of the IK solution on position level. Bocsi *et al.* suggested using structured output machine-learning methods for selecting a particular solution from a set of all possible solutions.[35] Similarly, Vannucci *et al.* proposed a system using growing neural gas for learning the global IK on position level.[36,37]

Practical implementations of these methods have been limited because of their complex formulations. Although a direct translation of these approaches can be realized for continuum robots, it must be realized that the kinematic formulation of continuum robots is fundamentally different from rigid robots. Unlike rigid robots, the kinematic model of continuum robots is dependent on environmental factors also. Nonetheless, the underlying scheme behind these methods can still be replicated and revised for our purpose.

This article presents a machine-learning-based approach for obtaining the global IK solution for a redundant continuum robot based on the author's previous work.[38,39] Developing on our previous work, an effective and unique technique for integrating feedback information about the tracking error in the learned system is proposed. The proposed method promises a simple, fast, and efficient learning of the global IK, even in the presence of singularities, without requiring any prior information about the robot and with few open parameters for the user to tune. In addition, the learning is done directly between the task space and actuator space, therefore requiring minimal sensor components and without requiring a representation for the configuration space. The generalizing ability of the regression algorithm (artificial neural networks, in our case) along with a simple IK formulation and sampling strategy leads to very less sample data requirement. The unique formulation of the IK learning problem allows the derived task space kinematic controller to display intelligent behavior even in an unstructured environment. Experimental results on a six DoF continuum manipulator indicate that the proposed approach is promising for continuum robot control and exhibits interesting behavior under external disturbances.

The next section describes the structure of learning algorithm for the IK solver and the underlying formulas that make the data-driven learning feasible. Further, analysis on a simulated nine DoF continuum manipulator is conducted to give empirical evidence of the efficacy of the feedback scheme and to showcase its effectiveness. A brief outlook into the experimental setup developed for a six DoF manipulator is then described, followed by characterization of the manipulator kinematics. Afterward, the training method and performance is described. Description of the experiments and their corresponding results are then mentioned in the next section. Finally, a summary of the article noting the advantages and disadvantages of the proposed method along with scope for improvements and possible developments in the future are stated in the last section.

Formulation of the Learning Architecture

The objective of this article is to develop kinematic controllers by developing models for the IK solutions of the continuum robot. The forward kinematics can be represented as a mapping between the actuator space (encoder value/length of cables, cable tension, pneumatic pressure, etc.), $\boldsymbol{q}$, and the end-effector coordinates, $\boldsymbol{x}$. Assuming that there are no environmental constraints, the forward kinematics can be represented by:

$$\boldsymbol{x} = f(\boldsymbol{q}) \quad (1),$$

where $\boldsymbol{x} \in \Re^m$ is the position and orientation vector; $\boldsymbol{q} \in \Re^n$ is the vector containing the actuator variables; and f is some surjective function. The IK model is a mapping between the end-effector coordinates and the actuator variables. Direct inversion of the forward function is not possible when $m < n$, that is, when the manipulator is redundant and the solution set of all possible solutions does not form a convex set. To simplify the inversion problem, the forward kinematics can be linearized at a point ($\boldsymbol{q}^o$), thereby obtaining the formulation:

$$\dot{\boldsymbol{x}} = J(\boldsymbol{q}^o)\ \dot{\boldsymbol{q}} \quad (2)$$

Here, J is the Jacobian matrix that transforms actuator velocities, $\dot{\boldsymbol{q}}$, to end-effector velocities, $\dot{\boldsymbol{x}}$. As shown by D'Souza *et al.*,[32] by spatially localizing the actuator variable $\boldsymbol{q}$, we can ensure convexity of the different IK problem, thereby making the learning problem tractable. By sacrificing slightly on the accuracy, the differential kinematics can be approximated as:

$$\Delta\boldsymbol{x} \approx J(\boldsymbol{q}^o)\Delta\boldsymbol{q} \quad (3)$$

$$J(\boldsymbol{q_i})\boldsymbol{q}_{i+1} \approx \boldsymbol{x}_{i+1} - f(\boldsymbol{q_i}) + J(\boldsymbol{q_i})\boldsymbol{q_i} \quad (4),$$

where $\boldsymbol{q}_{i+1}$ is the actuator configuration that archives the end-effector position $\boldsymbol{x}_{i+1}$, whereas $\boldsymbol{q_i}$ is the current actuator configuration. This not only allows us to learn the IK on a position level but also facilitates spatially localizing $\boldsymbol{q}$ by the sampling method rather than the learning architecture. The spatial localization can be done by ensuring that $|\boldsymbol{q}_{i+1} - \boldsymbol{q_i}|$ is bounded. Note that this will indirectly constraint the end-effector motion spatially. From empirical data, it is recommended that: $|\boldsymbol{q}_{i+1} - \boldsymbol{q_i}| < \epsilon$, where ϵ is between 3% and 10% of the actuator range. For faster exploration, a higher ϵ is better; however, a lower value provides better accuracy. Much lower values of ϵ can theoretically provide better accuracy, but, in reality, environmental noise will overpower the information present in the data.

The exploration strategy for collecting sample data involves continuous motor babbling (random actuator motion) while ensuring the spatial locality of the actuator variable. Now we can employ any universal function approximator to directly learn the mapping: $(\boldsymbol{x}_{i+1}, \boldsymbol{q_i}) \rightarrow (\boldsymbol{q}_{i+1})$. In our case, we are using a single hidden-layer Multi-Layer Perceptron for this purpose. We are using tan-sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. Since the data are naturally bounded by the actuator range and due to the tan-sigmoid transfer function and learning algorithm, the learned network will always output a valid joint configuration, thus ensuring valid motions. Now an important concern is what the response of the learned network would be when the target positions ($\boldsymbol{x}_{i+1}$) cannot be achieved from the current joint configuration by a local motion (since the network is trained with data that are obtained by local motions only). One can expect the learned network to be similar to the form given next:

$$\boldsymbol{q}_{i+1} = G(\boldsymbol{x}_{i+1} - f(\boldsymbol{q_i}) + J\boldsymbol{q_i}) \quad (5),$$

where G is a generalized inverse of $J(\boldsymbol{q_i})$. As noted in the authors' previous work,[38] when the target inputted is not a local point, the network response just gets scaled, as one would expect from Equation (5), thereby bringing the end-effector position closer to the target. Repeating the algorithm eventually leads to the network converging near the target position.

Since neural networks have the ability to generalize well, global learning of the IK can be done without exploring the complete actuator space, which is very large even for our six DoF manipulator. This is also aided by the fact that there is always a high correlation between local Jacobians. However, it is important that the exploration process obtains data from the boundaries of task space for maximal utilization of the workspace.

As stated earlier, eventually we intend to use these manipulators in complex environments and it is necessary for our controllers to exploit the adaptability of the physical body. Simple controllers that learn the mapping $(\boldsymbol{x}_{i+1},\ \boldsymbol{q_i}) \rightarrow (\boldsymbol{q}_{i+1})$ will fail in the presence of environmental constraints, because for continuum robots, the forward kinematic model is also dependent on the environment. This is due to the numerous underactuated and compliant joints present in a continuum robot. It is a hard task to model all the contacts and to get the subsequent kinematic model. Therefore, we propose a simpler way to incorporate feedback correction of errors occurring due to unstructured environmental factors and showcase that this simple strategy can perform well even without any modification of the learned network.

Consider a controller that learns the mapping regulated by a different formulation of Equation (6) shown next:

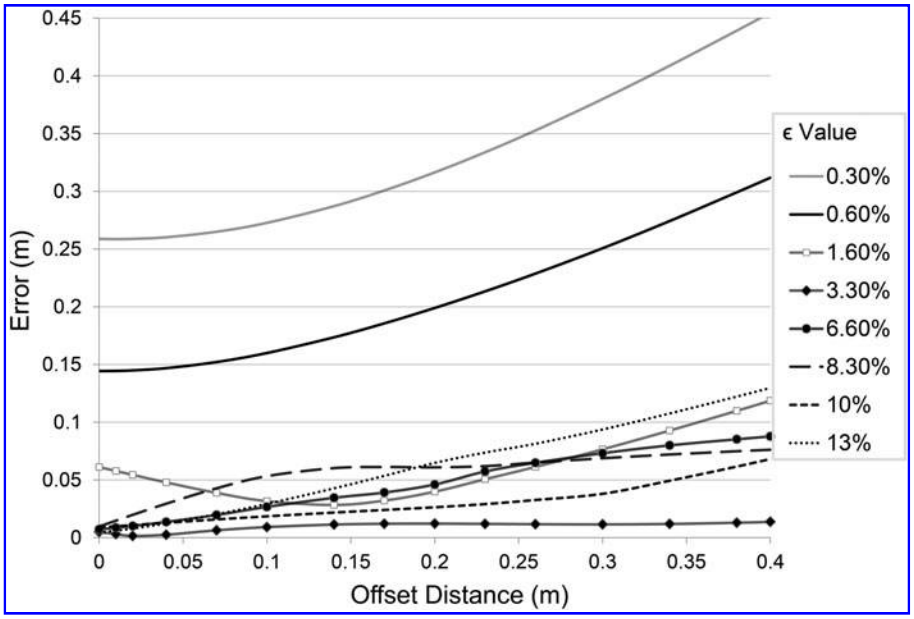

FIG. 1. Performance of the closed loop kinematic controller with offset added to kinematic model for different values of ϵ. The target is a fixed point for all cases.

$$q_{i+1} = G(x_{i+1} - x_i) + q_i \tag{6}$$

We can expect a network that learns the mapping: $(x_{i+1}, q_i, x_i) \rightarrow (q_{i+1})$ to always move along the direction of the Jacobian matrix scaled by the error in tracking: $(x_{i+1} - x_i)$. For the case of rigid robots, the information obtained from the end-effector position x_i is redundant; however, that is not the case for continuum robots. As the estimate of the Jacobian matrix by the learned network is based only on the current joint configuration, it will not be same as the actual Jacobian matrix. However, we argue that even this inaccurate estimate of the Jacobian matrix is enough to force the motion of the end effector in following a path of minimum possible error under external constraints. In other words, we claim that in the presence of obstacles a decent strategy involves trying to reach the target toward the current estimate of the Jacobian matrix and the high dimensionality and compliance of the body will guide the manipulator in the best path, as long as the controller tries to reduce the tracking error with the current estimate of the Jacobian matrix. In addition, adding the end-effector position as an input makes the learning more tolerant to noise incurred due to the stochasticity of the system, which greatly improves the learning process. Since it is hard to predict the underlying algorithm embedded in the network after learning, testing and validation of the proposed learning scheme under extreme situations is done with the help of simulations in the next section.

Simulations

In this section, two simulations are performed to test and validate the scheme for integrating the end-effector feedback to the learning scheme. Through these simulations, we try to demonstrate that the proposed controller can accommodate changes in kinematics of the robot without retraining the network. The simulations are conducted on a kinematic model of the bionic handling assistant.[40] The continuum kinematics is modeled by a constant curvature approximation, and the manipulator has nine DoFs.

To begin with, data samples are generated by continuous motor babbling, as mentioned in the previous section, and the mapping $(x_{i+1}, q_i, x_i) \rightarrow (q_{i+1})$ is learned. During the learning phase, no external disturbances are applied to the system, and, therefore, information about the current end-effector position (x_i) is actually redundant. Our interest lies in studying the effects of external factors on the manipulator and the subsequent response of the learned system.

Since the simulation model is purely kinematic, it is not possible to apply external forces directly. Therefore, the only way to model external factors is to modify the forward kinematics of the model itself. First, we set an offset to the end-effector position outputted by the simulator and observe the performance of the solver. This is equivalent to adding a constant value to the forward kinematic model. Note that the

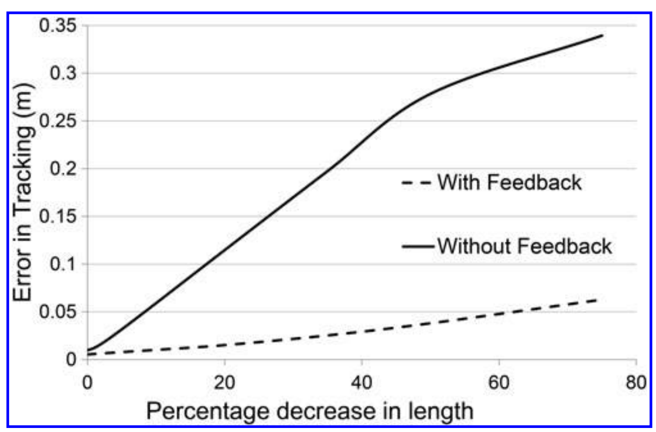

FIG. 2. Comparison of the performance of the closed loop kinematic controller and the open loop kinematic controller with non-linear changes in the forward kinematics. The target is a fixed point for all cases.

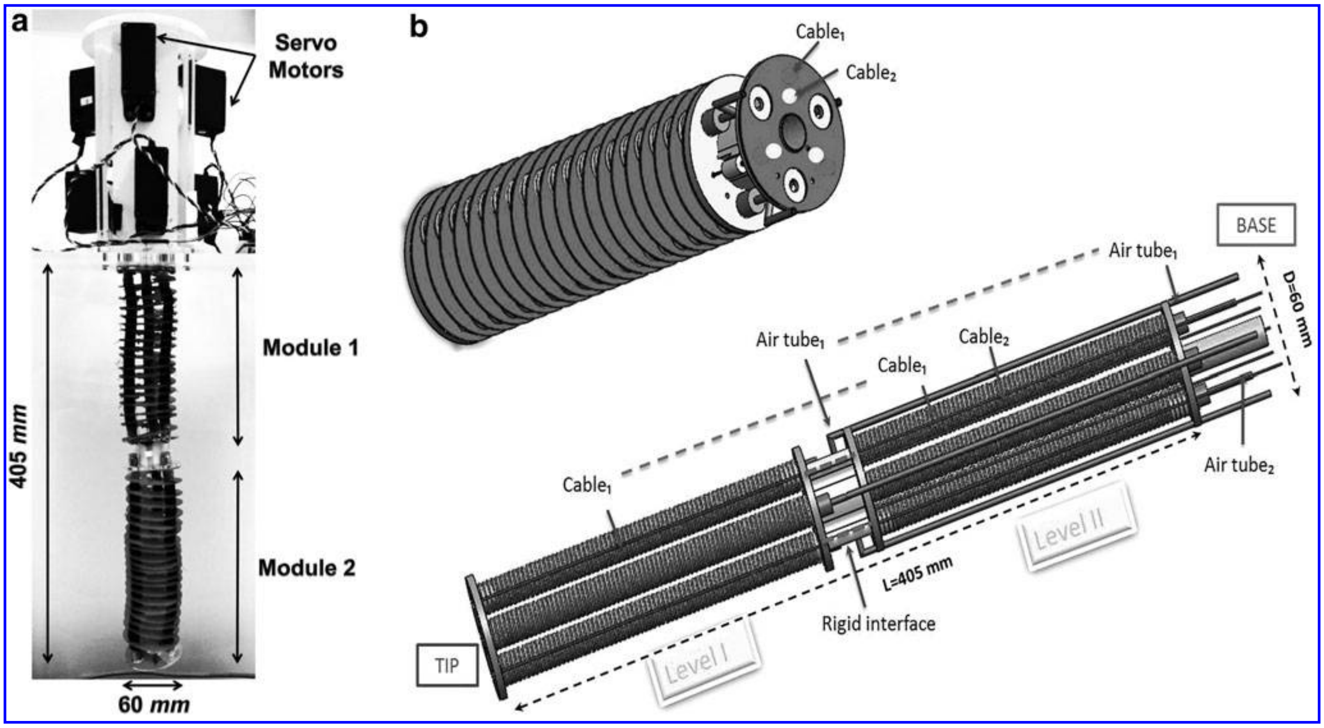

FIG. 3. **(a)** The six DoF continuum manipulator. **(b)** CAD model of the design. DoF, degree of freedom.

offset is adding after learning, and the IK solver does not have any information about the offset other than what is observed from the end-effector position. The target position is fixed in all cases. Therefore, the solver has to output different actuator configurations for reaching the same target because of the offset.

Figure 1 shows the error in tracking by the solver, with the change in offset value added in the X direction. The ϵ value is shown in percentage of the actuator range to showcase its effect on efficient learning. The number of samples collected is the same for all experiments. We can observe that for an optimum value of ϵ, the error in tracking can be reduced by the proposed feedback scheme. If there were to be no end-effector feedback in the IK solver, the tracking error will be directly proportional to the offset distance as the end effector will always depict the same actuator configuration, since the inputs (target) remain the same.

Furthermore, we can realize more nonlinear variations in the forward model by modifying the cable lengths after learning the IK model. Figure 2 shows the performance of the IK solver with end-effector feedback and the IK solver without feedback when the initial length of one of the nine cables is reduced. For more clarity, if the current cable length is q^o, the modified forward model always calculates the end-effector position with the new cable length $q^o - K$ (K is a constant and is denoted with respect to the initial cable length in Fig. 2); whereas for the learned IK solver, the input still remains as q^o. For the case of the IK solver with end-effector

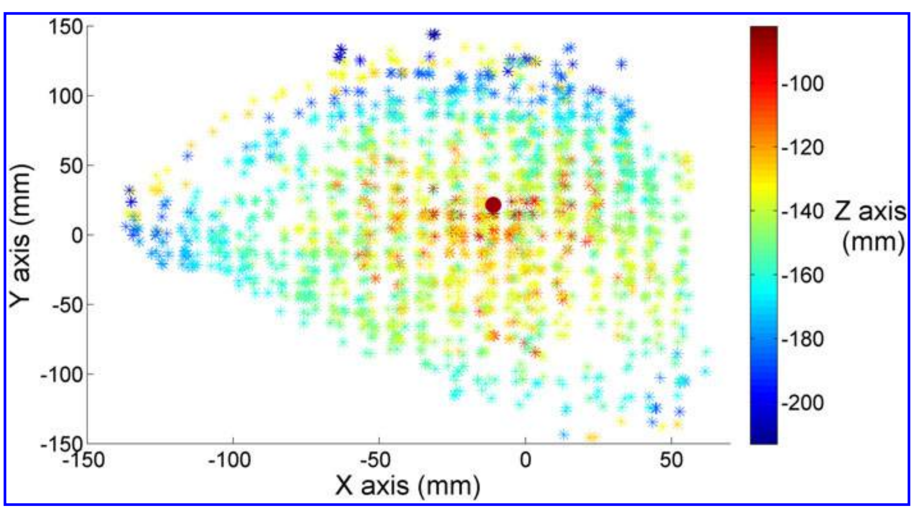

FIG. 4. Three-dimensional workspace of the manipulator. Color images available online at www.liebertpub.com/soro

feedback, the position of the end effector along with the information learned from the initial kinematic model is enough for accurate tracking of the target. It must be brought to the attention of the reader that the network is never explicitly directed to reduce the tracking error; the underlying error correction system arises purely from the data and the learning formulation.

Experimental Setup

For testing and validating the proposed learning controllers, we use a recently developed modular continuum manipulator designed for showering application.[41] Since the manipulator is developed for service applications, it is very important that the manipulator is inherently safe whereas the controller is reasonably accurate.

The setup is composed of two hybrid modules, with each module having three pneumatic and three tendon-driven actuators. The McKibben-based flexible fluidic actuators are to be used in tandem with the inelastic cable-driven actuators for extension, compression, and stiffening. The actuators are supported externally by a flexible helicoidal structure that has been inserted along the entire module, thus providing appropriate structural rigidity for our application while maintaining the dexterity of the arm. For our experiments, we will use only cable-driven actuation. Hence, the manipulator has only six active DoF. The unactuated length of the manipulator is 40.5 cm and its diameter is 6 cm.

The cables are actuated by six HS-785HB Winch Servo Motors. Servo motors are used for their ease of control in the actuator space; however, they do cause undesirable jerky motion. For tracking the position and orientation of the manipulator, the Aurora® tracking system (Northern Digital, Inc.) is used with a six DoF electromagnetic probe. The probe is attached at the end of the manipulator. If the environment is free of electromagnetic disturbances, the system specifies an accuracy of 0.70 mm and 0.30° (RMS). The Aurora system also specifies the uncertainty of each measurement, which is useful during the learning step for removing outliers in the data. The manipulator setup for the experiments is shown in Figure 3.

Manipulator Characterization

In this section, a brief analysis of the manipulator characteristics is summarized. A representation of the manipulator workspace is shown in Figure 4, which is obtained by motor babbling (The range of orientation is 52°, 135°, 128°). The un-actuated home position is marked with a larger circle. Two thousand sample points were measured for this purpose. The same data will be used for the learning process, which is described in the next section.

From a direct observation of the workspace, we can notice that the workspace is skewed. This is because of the asymmetric geometry of the manipulator, which arises from many factors. This is further indication of the need for model-free learning approaches in these scenarios. Also, few outliers can be observed near the boundaries as the probe reaches the boundary of the electromagnetic field. Therefore, during learning, each observation is prioritized with respect to the uncertainty in tracking, which is provided by the Aurora system. This will help reduce the effects of noise in the tracking system.

Table 1. Residuals of the Repeatability Test

	Standard deviation	*Mean and standard deviation of absolute residuals*
X (mm)	5.2	4.4 ± 2.9
Y (mm)	3.9	3.1 ± 2.3
Z (mm)	2.5	1.9 ± 1.7
Yaw (degrees)	1.2	0.8 ± 0.9
Pitch (degrees)	2.4	1.9 ± 1.5
Roll (degrees)	2.8	2.3 ± 1.7

Currently, the main drawback of the manipulator design is the slacking of the cables due to the parallel configuration of the cables along the modules. This not only leads to kinematic singularities that reduce the performance of the IK solver but also condenses the reachable workspace of the manipulator. The singular configurations do not interrupt the learning process as they can be considered as the non-homogenous solutions to the non-homogenous Equation (6).

In addition, the stochasticity of the manipulator kinematics is analyzed to provide an indication of the best achievable accuracy of the inverse model without any feedback. This is done by first recording the coordinates of the end effector for 100 random joint configurations, repeating this step four times to get the variability in the data, calculating the mean value of the end-effector position for each 100 joint configurations, and calculating the residuals of each observation with respect to its corresponding mean. The standard deviation for all 400 residuals along with the mean of the absolute residuals is shown in Table 1. This can be considered the expected performance of the best possible IK solver without any feedback correction, provided that the learning is perfect with infinite sample data. Similar trends can be observed in the experimental results too, but with the aid of the feedback controller.

Learning Overview

A multilayer perceptron with a single hidden layer is used for learning the mapping: $(\boldsymbol{x}_{i+1},\ \boldsymbol{q}_i, \boldsymbol{x}_i) \rightarrow (\boldsymbol{q}_{i+1})$. Tan-sigmoid transfer function is employed in the hidden layer, and a linear transfer function is employed in the output layer. For the implementation on the real manipulator, we learn two IK solvers: one only for position and one for both position and orientation (Euler angles). Therefore, we have one network with 12 inputs and another with 18 inputs. Both networks output the six motor positions. Bayesian regularization backpropagation algorithm is used for training the network. Two thousand samples are collected for training and divided in the ratio 70:30 for training and testing. The ϵ value used for

Table 2. Training Performance

	IK solver for position	*IK solver for pose*
Layer size	30	30
Epochs	61	138
Training NMSE	0.0086	0.0074
Testing NMSE	0.0099	0.0092
Training time (s)	31	100

IK, inverse kinematics; NMSE, normalized mean squared error.

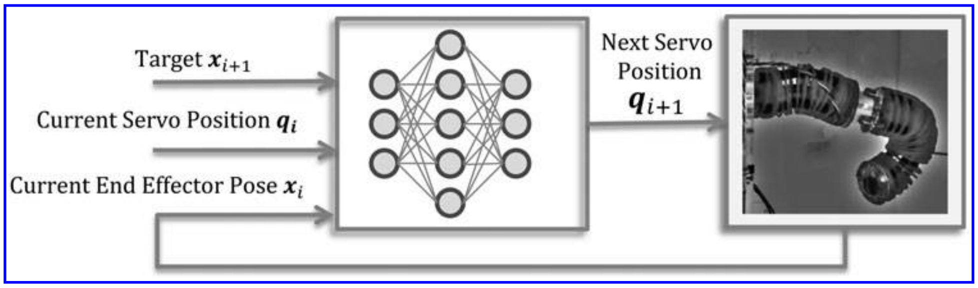

FIG. 5. Overview of the closed loop kinematic controller.

the exploration is 5% of the total actuator range. Each sample is collected after a delay of 2 s to ensure that the manipulator comes to a complete rest. The uncertainty in each observation is used as a scaling factor in calculating the mean squared errors for reducing the effect of noise in the training process. Table 2 shows the training performance for each IK solver.

In the whole learning process mentioned, there are only two open parameters for the user to decide. One is the exploration constant, ϵ, which is decided by the accuracy requirements and limited by the stochasticity of the manipulator. Generally, a value between 3% and 10% of the actuator range should guarantee valid learning. In addition, it is always possible to generate sample points for a higher ϵ by using data points of a lower ϵ by sacrificing on the number of data points. The second open parameter is size of the neural network, which is easier to determine.

Experiments and Results

To test and validate the performance of the solver, five sets of experiments are conducted for real-time kinematic control of the manipulator with the two learned IK solvers. The experiments are conducted for evaluating the performance of the solver for global point-to-point motion, local path following, local path following in an unstructured environment, and disturbance rejection while maintaining a particular position. For all the experiments, the manipulator starts from the home position, which is the configuration with all the actuators at the zero position. Figure 5 shows an overview of the IK solver-based control system. The weights of the neural network are also not updated during any of the experiments.

Point-to-point motion for pose control

For the first experiment, 25 random points are selected from the manipulator workspace and the manipulator is commanded to reach each pose in succession by using the solution provided by the IK solver. The points are selected from the workspace to ensure that the target is reachable as it is difficult to know beforehand whether a particular position and orientation is reachable. Since the targets are not local points, the solver will require few steps to converge. This makes the solutions fundamentally different from the training outputs and dependent on the initial conditions. Therefore, it is possible to obtain multiple solutions for the same target in case the manipulator is redundant for that case. The IK solver converges near the target within ten steps for all the targets. There is a delay of 1 s between each step, during execution, to ensure that the static state is reached. Table 3 shows the average and standard deviations of the absolute errors incurred while tracking the targets.

Table 3. Point-to-Point Motion Performance

	Mean error	*Standard deviation*
Position (mm)	9.67	5.33
X (mm)	5.53	4.08
Y (mm)	5.04	4.59
Z (mm)	4.03	3.56
Yaw (degrees)	2.76	5.42
Pitch (degrees)	1.84	1.83
Roll (degrees)	3.85	7.02

Trajectory following with IK solver for position and pose

The next experiment involves testing the performance of the solver in following a continuous path in the task space. Although the targets are local points, the difficulty lies in the fact that the corresponding actuator configurations need not be continuous in the actuator space. There are two trials conducted for this experiment. The first trial is conducted with the IK solver for position, with the target path being a straight line of length 20 cm divided into 40 waypoints. The IK solver is given only one step to provide the solution for each waypoint, with each step given 1 s to settle. The second trial is conducted with the IK solver for pose, with the same target positions and a fixed orientation of 90°, 0°, and 180°.

Figure 6 shows the errors in position, and Figures 7 and 8 show the errors in orientation of the manipulator in following the trajectory. The tracking error shown in Figure 7 is just for comparison purpose with the controller for pose. The average positional error for line following using the kinematic

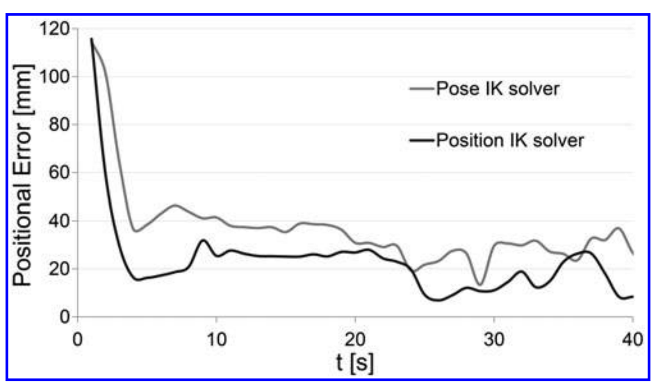

FIG. 6. Positional error in tracking for the two kinematic controllers.

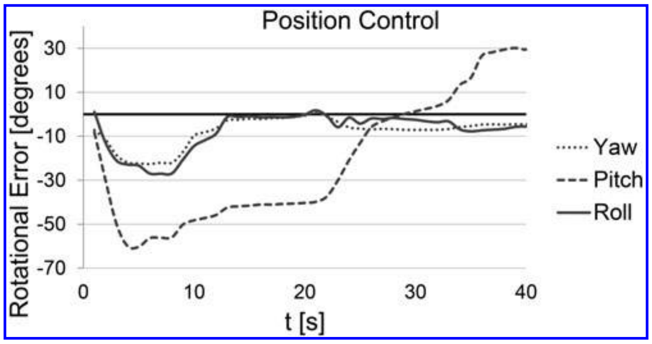

FIG. 7. Orientation error for the position kinematic controller given for evaluating the kinematic controller for pose.

controller for position is 23 mm, and the same for the pose controller is 37 mm. Since the target is updated every step, the IK solver would not be able to converge to the best solution for each target, and, consequently, the errors are higher for this experiment. In case of trajectory following with a fixed orientation, it must be mentioned that some of the targets are statically unreachable. It is in our interest to analyze the IK solver even if the targets are not reachable.

Since the IK model is constructed by a neural network, the behavior of the solver when the target is an outlier (for the case when the target is not reachable) is difficult to be foreseen. The underlying algorithm is majorly dependent on the training data. Figure 9a shows the configurations of the manipulator during the path following task using the IK solver for position. Due to the manipulator constraints, it is impossible for the IK solver for pose to reach the desired target during the beginning of the task and toward the end. The higher errors in position as shown in Figure 6 can be attributed to this. We can see that there is a trade-off between tracking the position and orientation of the robot. However, presently, it is difficult to quantify how the solver prioritizes for reducing the orientation and position tracking errors or even whether there is such a trade-off internally.

Trajectory following in an unstructured environment

The next test involves seeing the effect of an unstructured environment while executing a task. The trajectory is the same straight line position path mentioned in the previous

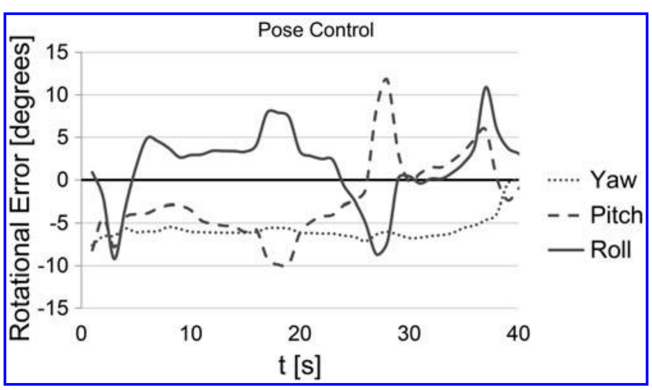

FIG. 8. Orientation error while tracking using the kinematic controller for pose.

section but with just 20 points instead of 40. Each step is updated at 0.25 s. This is to ensure that some of the momentum is still conserved between each step. A cylindrical obstacle of diameter 8 cm is placed in between the path such that the center of the cylinder lies around 10–30 (Fig. 10). We are only analyzing the performance of the IK solver for position as the manipulator is only redundant for this task. Figure 9b shows the configurations of the manipulator while performing the task. The numerous DoFs and compliance in the body aid the controller in tracking the best possible path under the circumstances.

By a quick comparison with the task shown in Figure 9a, we can see the various kinematic configurations that the manipulator is able to achieve with the help of the environment, which are otherwise unattainable. This is one of the key characteristics of continuum robots. The ability of continuum robots to have modified kinematics in the presence of external forces makes it almost impossible to get accurate formulation of the kinematics; on the contrary, it opens up the possibility of using the environment to guide the manipulator and to facilitate the control objective.

Through this experiment, we demonstrate that approximate IK models with end-effector feedback can work in collaboration with the morphological properties of the robot to accomplish tasks even in an unstructured environment. Also noteworthy is the importance of friction in this task. We noticed that very low friction can lead to slippage and cause the manipulator to overshoot from its desired target. Figure 10 shows the top view of the path that the end-effector traces in the presence of the obstacle, and Figure 11 shows the path without the obstacle. The rigidity of the helicoidal structure does prevent the manipulator from fully conforming to the shape of the obstacle, and the tracking error experienced due to that is visible toward the end of the path in Figures 9b and 10. The forces on the manipulator due to the height of the obstacle also affect the tracking performance.

Disturbance rejection during position control

The final test involves showcasing the importance of the end-effector feedback for compensating external disturbances. Under external disturbances, the kinematics of the manipulator will get modified, as pointed out in the Simulations section. Therefore, the IK solver needs to provide new solutions for reaching the same target. For this experiment, the external forces are applied by hand after the manipulator has converged to the desired target. Due to the compliance of the body, the manipulator can deform without disturbing the servo position. Consequently, the only change in the input to the IK solver is the change in the end-effector position.

We are not concerned with the exact forces or deformations applied to the manipulator as the experiments are aimed at seeing whether the IK solver can output different solutions just by the change in end-effector position and whether the solutions can decrease the error in tracking. Clearly, there would not be any reaction from an IK solver without the feedback about the end-effector position as the actuator configuration remains the same under the disturbance. Besides that, due to the change in kinematics induced by the disturbance, the new inputs provided to the neural network are outliers when compared with the training data. Thus, it is essential to verify that the network can provide decent

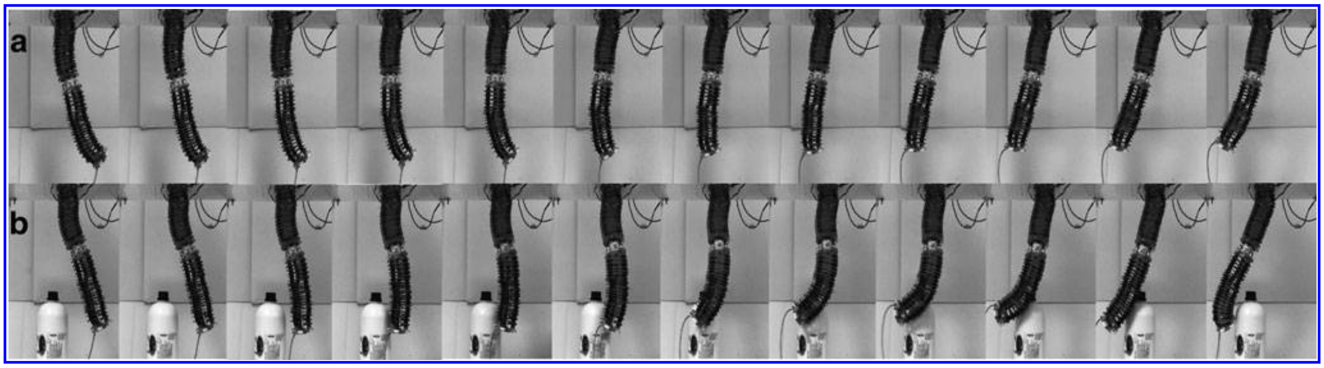

FIG. 9. **(a)** Configuration of the real manipulator at different time steps during the line following task. **(b)** Configuration of the real manipulator for the same corresponding time steps during the line following task in the presence of an obstacle.

solutions and more importantly, to see that the network does not get saturated.

Figure 12 shows the configuration of the manipulator after disturbance and the final configuration derived from the IK solver for four different cases. The four trials are conducted in sequence, and the solver is updated every 0.5 s. From the figures, we can observe that the controller is able to respond exceedingly well to the external disturbances. The error in tracking is reduced even with the change in kinematics; from the images, we can observe that the manipulator can reach the same target with different configurations. Again, it must be emphasized that the IK solver is never explicitly commanded to reduce the tracking error and also that the mechanical constraints of the manipulator hamper further reduction of the tracking error. The perceived controller arises purely from the IK representation, and the sample data.

Conclusion and Future Work

In conclusion, this article introduces a machine-learning-based kinematic controller for continuum robots. Due to the fact that accurate and fast analytical models are difficult to implement for continuum robots, model-free methods can perform just as well. Not only is the proposed approach applicable to any form of continuum robot, but it also has very few parameters to tune and very less sample data requirements compared with existing methods (Like Neural Gas). For instance, for the six DoF manipulator, we required just 2 h to generate the data, learn the network, and implement the controller. We consider it very important to have controllers with minimal sensory requirements to make its implementation practical and also comparable with rigid robots. In spite of many approximations and sensory handicaps, the proposed IK solver can generate multiple global solutions to the IK problem through an iterative method. Coupled with a unique IK formulation that allows for a feedback mechanism, the kinematic controller can perform highly adaptive motion even in a completely unstructured environment. The uniqueness of the proposed controller is that in some sense the feedback is used for modifying the kinematic model, thereby helping the manipulator adapt in an unstructured environment, and this is unlike any formulation possible in rigid robots. To the best of our knowledge, this methodology has not been implemented on continuum robots either.

In retrospect, there are a few design modifications that we believe can improve the performance of the controller. The

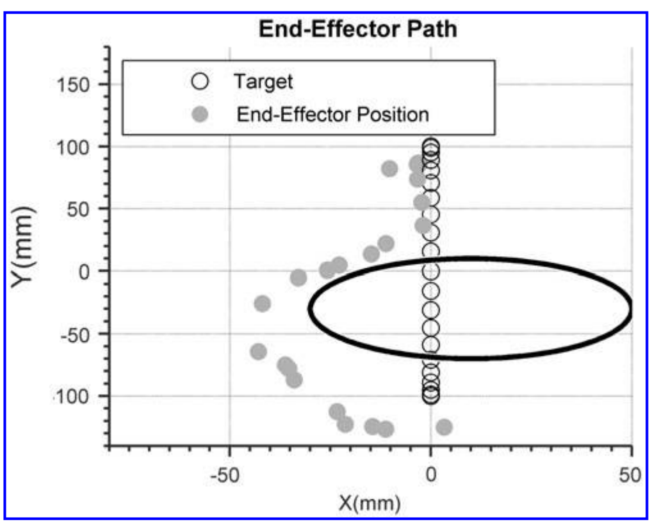

FIG. 10. Path of the end effector in the presence of an obstacle in a line following task.

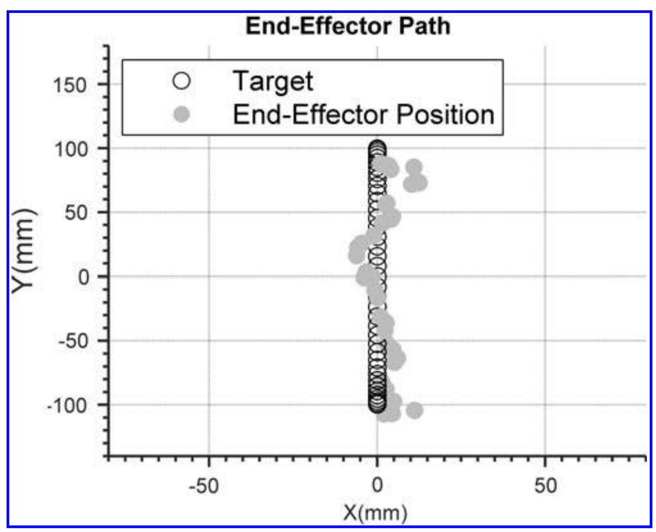

FIG. 11. Path of the end effector in an uninterrupted line following task.

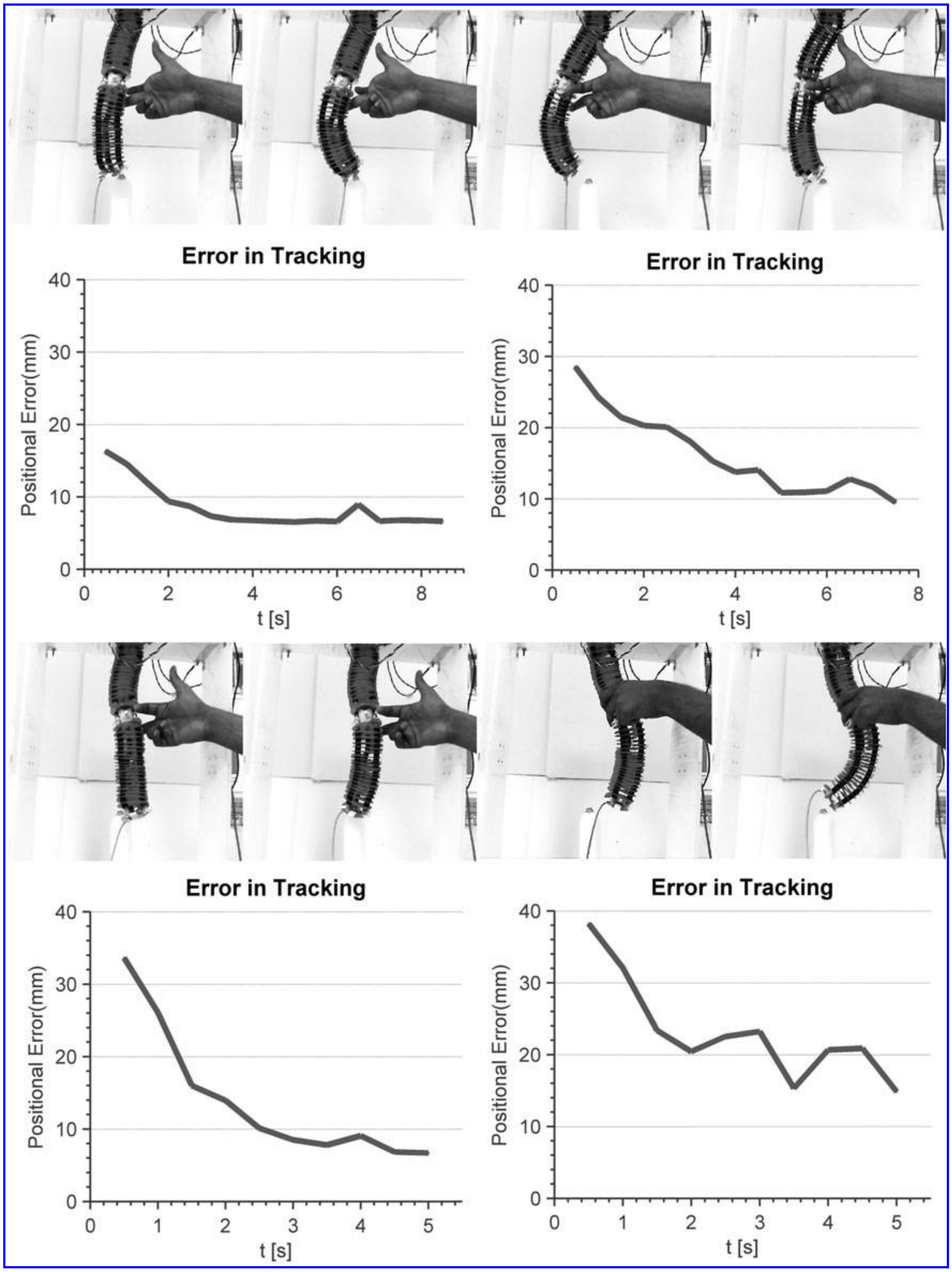

FIG. 12. Disturbance rejection using the kinematic controller. Four cases are shown in the experiment with configuration of the manipulator after disturbance shown first and the final configuration shown in the end. The complete tracking error is shown below.

kinematic controller proposed in this article is implemented by using high-gain servo motors. Although this simplifies the low-level control of cable lengths, such a system completely nullifies the dynamics of the continuum manipulator. This also makes the control system highly inefficient in terms of energy consumption. One of the potential improvements in the future would be to have low-gain controllers in the actuator space, which could significantly improve the performance of the kinematic controller by providing smoother and faster motion. Another point of interest is the possibility of using a hybrid combination of force and stretch sensors for force and position control as proposed by Bajo and Simaan[42] and developing controllers that consider the dynamics of the manipulator too in motion planning.

Acknowledgments

The authors would like to acknowledge the support by the European Commission through the I-SUPPORT project (HORIZON 2020 PHC-19, No. 643666). They would also like to thank the Italian Ministry of Foreign Affairs and International Cooperation DGSP-UST for the support through the Joint Laboratory on Biorobotics Engineering project.

Author Disclosure Statement

No competing financial interests exist.

References

1. Laschi C, Cianchetti M. Soft robotics: New perspectives for robot bodyware and control. Front Bioeng Biotechnol 2014;2:1–5.
2. Pfeifer R, Lungarella M, Iida F. Self-organization, embodiment, and biologically inspired robotics. Science 2007; 318:1088–1093.
3. Brown E, Rodenberg N, Amend J, *et al.* Universal robotic gripper based on the jamming of granular material. Proc Natl Acad Sci U S A 2010;107:18809–18814.
4. Manti M, Hassan T, Passetti G, D'Elia N, Laschi C, Cianchetti M. A bioinspired soft robotic gripper for adaptable and effective grasping. Soft Robot 2015;2:107–116.
5. Lin HT, Liesk GG, Trimmer B. GoQBot: A caterpillar inspired soft-bodied rolling robot. Bioinspir Biomim 2011; 6:026007.
6. Kim S, Spenko M, Trujillo S, Heyneman B, Mattoli V, Cutkosky MR. Whole body adhesion: Hierarchical, directional and distributed control of adhesive forces for a climbing robot. *IEEE International Conference on Robotics and Automation.* Rome, Italy, April 2007, pp. 1268–1273.
7. Ijspeert AJ, Crespi A, Ryczko D, Cabelguen JM. From swimming to walking with a salamander robot driven by a spinal cord model. Science 2007;315:1416.
8. Marchese AD, Onal CD, Rus D. Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators. Soft Robot 2014;1:75–87.
9. Calisti M, Falotico E, Laschi C. Hopping on uneven terrains with an underwater one-legged robot. IEEE Robot Autom Lett 2016;1:461–468.
10. Polygerinos P, Lyne S, Wang Z, Nicolini LF, Mosadegh B, Whitesides GM, *et al.* Towards a soft pneumatic glove for hand rehabilitation. *IEEE International Conference on Intelligent Robots and Systems*. Tokyo, Japan, 2013, pp. 1512–1517.
11. Calisti M, Giorelli M, Levy G, Mazzolai B, Hochner B, Laschi C, *et al.* An octopus-bioinspired solution to movement and manipulation for soft robots. Bioinspir Biomim 2011;6:036002.
12. Webster R, Jones B. Design and kinematic modeling of constant curvature continuum robots: A review. Int J Robot Res 2010;29:1661–1683.
13. Renda F, Cianchetti M, Giorelli M, Arienti A, Laschi C. A 3D steady-state model of a tendon-driven continuum soft manipulator inspired by the octopus arm. Bioinspir Biomim 2012;7:025006.
14. Renda F, Giorelli M, Calisti M, Cianchetti M, Laschi C. Dynamic model of a multibending soft robot arm driven by cables. IEEE Trans Robot 2014;30:1109–1122.
15. Zambrano D, Cianchetti M, Laschi C. The morphological computation principles as a new paradigm for robotic design. In: Hauser H, Füchslin RM, Pfeifer R. (Eds.) Opinions and Outlooks on Morphological Computation 2014, pp. 214–225.
16. Camarillo D, Carlson C, Salisbury J. Task-space control of continuum manipulators with coupled tendon drive. Exp Robot 2009;54:271–280.
17. Kapadia A, Walker I. Task-space control of extensible continuum manipulators. *IEEE/RSJ International Conference on Intelligent Robots and Systems.* San Francisco, CA, 2011, pp. 1087–1092.
18. Kapadia A, Fry K, Walker I. Empirical investigation of closed-loop control of extensible continuum manipulators. *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Chicago, IL, 2014, pp. 329–335.
19. Penning R, Jung J, Borgstadt J, Ferrier N, Zinn M. Towards closed loop control of a continuum robotic manipulator for medical applications. *IEEE International Conference on Robotics and Automation*. Shanghai, 2011, pp. 4822–4827.
20. Penning R, Jung J, Ferrier N, Zinn M. An evaluation of closed-loop control options for continuum manipulators. *IEEE International Conference on Robotics and Automation.* Saint Paul, MN, 2012, pp. 5392–5397.
21. Marchese A, Komorowski K, Onal C, Rus D. Design and control of a soft and continuously deformable 2D robotic manipulation system. *IEEE International Conference on Robotics and Automation*. Hong Kong, 2014, pp. 2189–2196.
22. Marchese A, Katzschmann R, Rus D. Whole arm planning for a soft and highly compliant 2D robotic manipulator. *IEEE/RSJ International Conference on Intelligent Robots and Systems.* Chicago, IL, 2014, pp. 554–560.
23. Duriez C. Control of elastic soft robots based on real-time finite element method. *IEEE International Conference on Robotics and Automation.* Karlsruhe, Germany, 2013, pp. 3982–3987.
24. Falkenhahn V, Hildebrandt A, Neumann R, Sawodny O. Dynamic control of the bionic handling assistant. *IEEE/ASME Transactions on Mechatronics*, 2017;22:6–17.
25. Yip M, Camarillo D. Model-less feedback control of continuum manipulators in constrained environments. IEEE Trans Robot 2014;30:880–889.
26. Yip M, Camarillo D. Model-less hybrid position/force control: A minimalist approach for continuum manipulators in unknown, constrained environments. IEEE Robot Autom Lett 2016;1:844–851.
27. Braganza D, Dawson D, Walker I, Nath N. A neural network controller for continuum robots. IEEE Trans Robot 2007;23:1270–1277.
28. Giorelli M, Renda F, Calisti M, Arienti A, Ferri G, Laschi C. Neural network and Jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature. IEEE Trans Robot 2015;31:823–834.
29. Rolf M, Steil J, Gienger M. Online Goal Babbling for rapid bootstrapping of inverse models in high dimensions. *IEEE International Conference on Development and Learning*. Frankfurt am Main, Germany, 2011, pp. 1–8.
30. Ansari Y, Falotico E, Mollard Y, Busch B, Cianchetti M, Laschi C. A Multiagent reinforcement learning approach for inverse kinematics of high dimensional manipulators with precision positioning. *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and*

Biomechatronics. Singapore, July 2016, art. no. 7523669, pp. 457–463.
31. Melingui A, Lakhal O, Daachi B, Mbede J, Merzouki R. Adaptive neural network control of a compact bionic handling arm. IEEE ASME T Mech 2015;20:2862–2875.
32. D'Souza A, Vijayakumar S, Schaal S. Learning inverse kinematics. *IEEE/RSJ International Conference on Intelligent Robots and Systems.* Maui, HI, 2001, pp. 298–303.
33. Jordan M, Rumelhart D. Forward models: Supervised learning with a distal teacher. Cogn Sci 1992;16:307–354.
34. Oyama E, Chong NY, Agah A, Maeda T. Inverse kinematics learning by modular architecture neural networks with performance prediction networks. *IEEE International Conference on Robotics and Automation.* Seoul, South Korea, 2001, pp. 1006–1012.
35. Bocsi B, Nguyen-Tuong D, Csato L, Scholkopf B, Peters J. Learning inverse kinematics with structured prediction. *IEEE/RSJ International Conference on Intelligent Robots and Systems.* San Francisco, CA, 2011, pp. 698–703.
36. Vannucci L, Cauli N, Falotico E, Bernardino A, Laschi C. Adaptive visual pursuit involving eye-head coordination and prediction of the target motion. *14th IEEERAS International Conference on Humanoid Robot.* Madrid, Spain, 2014, pp. 541–546.
37. Vannucci L, Falotico E, Di Lecce N, Dario P, Laschi C. 2015. Integrating feedback and predictive control in a Bio-inspired model of visual pursuit implemented on a humanoid robot. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9222, pp. 256–267.
38. Thuruthel T, Falotico E, Cianchetti M, Laschi C. Learning global inverse kinematics solutions for a continuum robot. *ROMANSY 21 - Robot Design, Dynamics and Control: Proceedings of the 21st CISM-IFToMM Symposium,* vol. 569. Udine, Italy: Springer International Publishing, June 20–23, 2016, pp. 47–54.
39. Thuruthel T, Falotico E, Cianchetti M, Renda F, Laschi C. Learning global inverse statics solution for a redundant soft robot. *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics.* Lisbon, Portugal, ICINCO, pp. 303–310.
40. Rolf M, Steil J. Constant curvature continuum kinematics as fast approximate model for the Bionic Handling Assistant. *IEEE/RSJ International Conference on Intelligent Robots and Systems.* Vilamoura, Portugal, 2012, pp. 3440–3446.
41. Manti M, Pratesi A, Falotico E, Cianchetti M, Laschi C. Soft assistive robot for personal care of elderly people. *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics.* Singapore, July 2016, art. no. 7523731, pp. 833–838.
42. Bajo A, Simaan N. Hybrid motion/force control of multibackbone continuum robots. Int J Robot Res 2015;35:422–434.

Address correspondence to:
Thomas George Thuruthel
The BioRobotics Institute
Scuola Superiore Sant'Anna
Pisa
Italy

E-mail: t.thuruthel@sssup.it

This article has been cited by:

1. George Thuruthel Thomas, Ansari Yasmin, Falotico Egidio, Laschi Cecilia. 2018. Control Strategies for Soft Robotic Manipulators: A Survey. *Soft Robotics* **5**:2, 149-163. [Abstract] [Full Text] [PDF] [PDF Plus]