

# Shape Estimation and Control of a Soft Continuum Robot Under External Payloads

Qingxiang Zhao , Graduate Student Member, IEEE, Jiewen Lai , Graduate Student Member, IEEE, Kaicheng Huang , Student Member, IEEE, Xiaobing Hu, and Henry K. Chu , Member, IEEE

**Abstract**—**Finding the shape of a continuum robot is a challenging task, especially in a constrained environment where external sensors like cameras are unable to provide sufficient information. In particular, accurate robot control becomes more difficult when the robot needs to interact with uncertain external payloads. In this article, we present the development of a soft robotic system with embedded sensors to help reconstruct the shape of the robot under different external disturbances. First, strain gauges were employed to perceive the current deformation state of the robot due to actuator inputs and external payloads. Then, a shape reconstruction (SR) method based on a spatial curve fitting approach was proposed, where multiple control points along the robot were predicted using Neural Networks. With the estimated shape, a local inverse kinematics model was developed so that the robot end-effector can be precisely controlled in quasi-static state to reach various positions, and position of end effector calculated via SR module was employed to improve the path following performance. Simulation and experiments were conducted and the results confirm that the proposed control system is effective to predict the shape and properly control the robot with or without external payloads.**

**Index Terms**—Continuum robot, inverse kinematics, neural networks (NNs), shape reconstruction (SR), uncertain external payloads.

## I. INTRODUCTION

THE CONCEPT of continuum robot, whose body can be freely deformed to interact with an object, was proposed by Robinson [1] in 1999. Together with other advantages such as miniature in size, hyper redundancy, and low exertion force, continuum robots have been attractive for use in surgery [2], inspection/repair [3], and self-exploring applications [4]. No

Manuscript received October 4, 2020; revised January 13, 2021 and May 27, 2021; accepted September 10, 2021. This work was supported in part by the Grant 4-RKDQ of The Hong Kong Polytechnic University. Recommended by Technical Editor A. N. Vargas and Senior Editor Gursel (SE) Alici. (Corresponding author: Henry K. Chu.)

Qingxiang Zhao, Jiewen Lai, Kaicheng Huang, and Henry K. Chu are with the Department of Mechanical Engineering, Hong Kong Polytechnic University, Hong Kong (e-mail: qingxiang.zhao@connect.polyu.hk; jw.lai@connect.polyu.hk; huangkc0907@hotmail.com; henry.chu@polyu.edu.hk).

Xiaobing Hu is with the School of Mechanical Engineering, Sichuan University, Chengdu 610017, Sichuan (e-mail: huxb@scu.edu.cn).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TMECH.2021.3116970>.

Digital Object Identifier 10.1109/TMECH.2021.3116970

matter for any operation, flexible continuum robots need to be controlled accurately. In addition, the shape of the flexible manipulator is anticipated to be visualized so that users can perform better handling. However, maneuvering and visualizing flexible manipulators in unstructured environments with high precision is challengable, as uncertain external payloads are unavoidable.

First, to deal with the visualization issue, some conventional solutions employed external sensors, such as stereo camera, and near-infrared systems [5], but they are not applicable in a constrained or occluded workspace. Alternatively, sensors can be embedded or integrated inside the manipulator. Scharff *et al.* [6] proposed a novel color-based 2-D shape measurement method, but external disturbances could distort the bellow. Similarly, deformation of elastomer can lead to length change in the chamber filled with liquid metal, like eutectic gallium and indium, resulting in the change of electronic resistance [7]. A novel soft sensor made by embedding conductive silicone into nonconductive silicone was proposed by Thomas *et al.* [8]. They succeeded sensing shape and estimating external force applied on the robot's tip. Venka-tasubramanian *et al.* [9] discretized a flexible robot as multiple serially connected pseudo rigid links, but solving the unknown angle of each joint to obtain the shape is computationally burdensome. Measuring the length of actuated chamber with linear Hall sensors is also feasible [10]. Electromagnetic (EM) sensors provide another way to measure the relative pose (distance) between a reference point (sensor) and the deformable body (magnet) [11]–[13]. Similar to Hall sensor, its performance could be interfered by strong magnetic fields. Different degrees of deformation can lead to strain change on a local area of robot, which inspired the use of strain/stress related sensors. For instance, fiber bragg gratings (FBG) [9], [14], [15] can sense the change in the curvature of an optical fiber by detecting the shift in the wavelength travelling in the fiber. The fiber can be very small in diameter and is immune to noise. FBG-related models generally regard the strain change of FBG is linearly elastic.

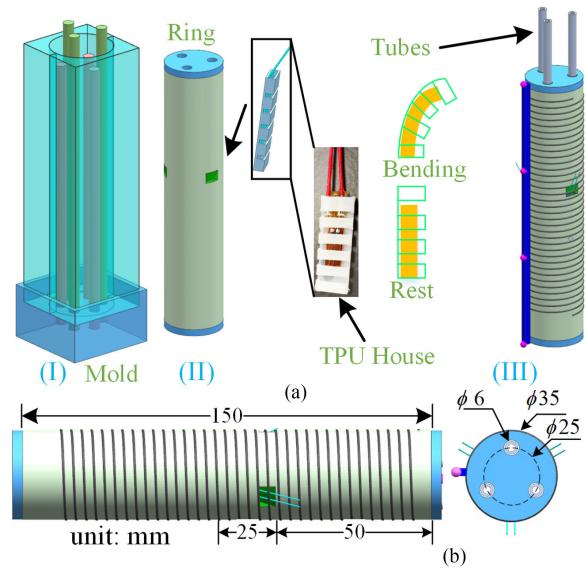
Like the linear elastic approximation applied with FBG sensors, it is equally important to build mathematical models to relate between the sensor readings and the manipulator shape. Song *et al.* [16], [17] fitted backbone using piecewise quadratic Bezier curve and the position of key points obtained by EM sensors; Inderjeet *et al.* [18] adopted Hodograph curve to fit 3-D backbone. Other related methods include: cubic spline [19], microarc interpolation [20], [21], and cubic Bezier curve [22].

However, most existing curve fitting algorithms only focused on 2-D curves, which did not consider the spatial shape of robot when it's influenced by external force, so the fundamental curve function should be a 3-D curve. In addition, Scimeca *et al.* [23] fabricated a capacitive tactile array that is mounted at the bottom of continuum robot. By sensing the force change of the sensors and using neural networks (NNs), they solved the shape in a plane manner. This is a simple solution, but “one to many” issue exists due to indirect measurement. Beam theory [24]–[26] regards flexible manipulator as a cantilever beam, and the spatial displacement of a manipulator was solved using internal payloads (from actuators) and external forces. This is theoretically feasible but quantifying every external force is indeed not practical.

Second, to control soft robot accurately under disturbance, versatile methods have provided solutions. The fabrication imperfections and external forces are essentially unavoidable, so fuzzy control theory with NNs [27] and with Jacobian-based method [28] are useful. To overcome the limitation of Jacobian matrix that singular position occurs for robots with infinite Degree of Freedoms (DoFs), Giorelli *et al.* [29] fused Jacobian and NNs, and solved the inverse kinematics of a cable-driven robot. Yip *et al.* [30] proposed an convex optimization algorithm to form feedback control in constrained environments. Online estimation was investigated by Fang in [31], where kinematics can be assumed as a local Gaussian model and updated continuously. Notably, some continuum robots [26], [32] can adjust the stiffness of robot to alleviate the influence of external payloads. Based on discrete-jointed robot formulation, researchers considered torque control for each virtual actuator [33], and energy-shaping approach to derive the control law [34]. Reinforcement learning provides a novel solution [35] to this challenge, where soft robot is commanded to learn the optimal control model, requiring little human intervention. It, however, needs long time to train the model. Li *et al.* [36] assumed the Jacobian matrix is constant within a local area, and estimated it by moving actuators with small increments and sensing the change of the end effector, which is time-consuming. Similarly, a novel strong tracking Kalman filter was proposed by Li *et al.* [37] to work adaptively in uncertain scenarios, in which a fading factor was introduced to accelerate the convergence and the covariance matrix was updated by an innovation sequence. Apart from them, model-based methods also contribute to trajectory tracking in dynamic environment [38]. Cosserat rod theory [39], virtual work principle [40], improved Piecewise Constant Curvature (PCC) algorithm [41], and stiffness of material [42], [43] are employed to find the mapping between forces (actuation inputs and external forces) and the pose of robot's end effector, which is an inverse process in control.

Precise control of a continuum robot with limited sensory information and unknown external payloads is a challenging task. To overcome it, this work is developed with the following contributions.

- 1) Provide proprioceptive shape sensing mechanism for visualizing multisegment continuum robot, where curve fitting process is divided into two subprocesses, ensuring time effectiveness and easing computation burden.



**Fig. 1.** Illustration of a single segment of continuum robot. (a) Detailed fabrication process. (b) Dimension of manipulator and location of sensors.

- 2) Consider the error of key points in fitting, proving the curve fitting method is robust.
- 3) Tip position feedback mechanism from SR is formed to the controller.
- 4) Adaptive step distance mechanism is embedded into the controller to adapt robot in varied disturbance.

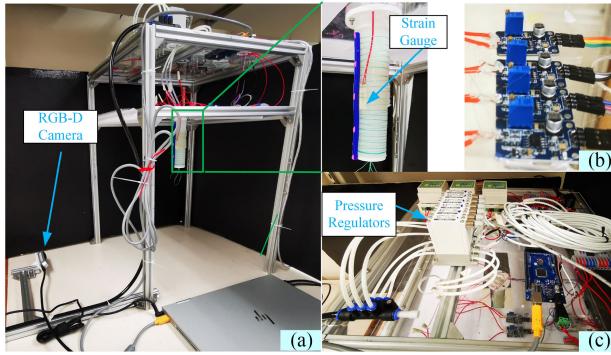
The rest of this article is organized as follows. Section II introduces the fabrication process of soft manipulator, where the performance of sensors is analyzed. Sections III and IV detail the shape reconstruction (SR) method and control scheme, respectively. Section V validates the proposed methods in simulation environment. Section VI examines the methods in testbed. Finally, Section VII concludes this article.

## II. EXPERIMENTAL SETUP

In this article, a pneumatic-driven continuum robot testbed was built to examine the performance of the proposed methods. The robot consists of two segments so that the end-effector can form different pose through different actuator inputs.

### A. Description of Testbed

Fig. 1(a) shows the fabrication process for a segment of the soft manipulator. First, silicone rubber was poured into a 3-D-printed mold [see Fig. 1(a)–(I)] and cured in an oven (45 °C). Then, two rings (Polylactic Acid or PLA material) were placed at the two ends, respectively, and an inextensible thread was attached at the center of the rings to limit the total length of one segment as 150 mm long [see Fig. 1(a)–(II)]. Before inserting three strain gauges into the grooves designed in the first step, each strain gauge was housed into a thermoplastic urethane (TPU) box so that during manipulator's deformation slippage between the sensor and the TPU house protects strain gauge from breakage (strain gauge can not be elongated significantly).



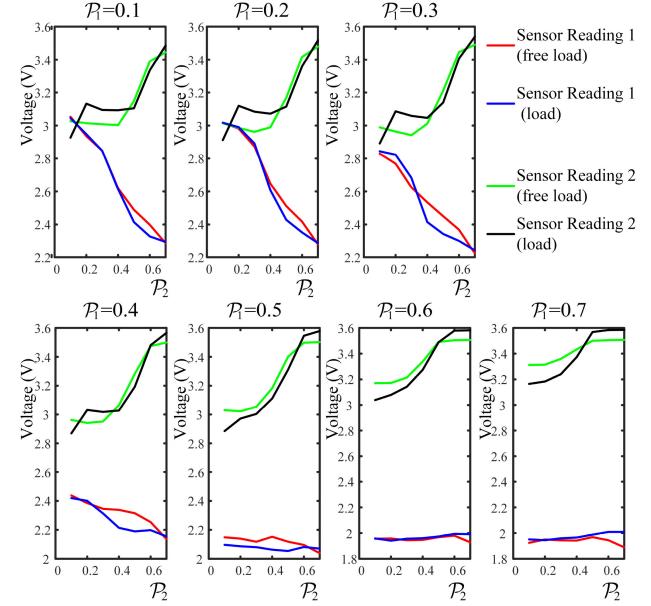
**Fig. 2.** (a) System setup with a single continuum robot. (b) Electrical resistance measurement module. (c) Hardware configuration.

Next, three tubes were attached at the chambers for actuation, and Nylon thread was twined around the soft manipulator to limit radial expansion upon being pressurized. To enable visual verification, pink markers were labelled along the body as the key control points and the intermediate points were labelled in blue marker [see Fig. 1(a)–(III)]. In addition, Fig. 1(b) showcases the dimension of a single-segment soft manipulator.

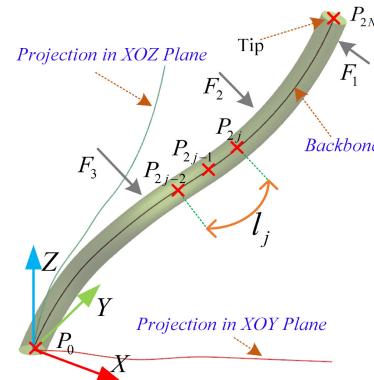
The pneumatic-driven manipulator was mounted on a aluminium frame, as shown in Fig. 2(a). The tubes were connected to pneumatic pressure regulators (SMC, ITV0030-BL). To evaluate the true position of the backbone, an RGB-D camera (Intel Realsense D415) was employed at a rate of 60 fps. The RGB-D camera was properly calibrated, which can detect the Cartesian positions of different markers as the ground truth. Readings from the strain gauges were collected via Arduino. Since strain gauges are sensitive to temperature, the room temperature is kept to 22.5°C, and a dummy strain gauge can also be added to the Wheatstone bridge to compensate the temperature effect. Fig. 2(c) shows the hardware configuration.

### B. Sensor Performance

A single-segment manipulator was used as an example to investigate the performance of bending sensors. We actuated two chambers by increasing the pressure of two chambers from 0 bar to 0.7 bar at an increment of 0.1 bar. Load-free and load conditions were compared to investigate sensors' performance. For the load scenario, an aluminium block (119g) was attached at the distal end of manipulator. Fig. 3 shows the comparison of the readings from two sensors. Increasing actuation pressure leads to a higher degree of deformation, and the sensor readings responded accordingly. The effect of the additional load on the sensor readings is obvious, which confirms that the sensors output a different sets of sensor readings. In addition, as shown in Fig. 3, both pressure inputs and sensor readings can provide useful information to evaluate the shape of the manipulator, but the sensor readings are unique to the load status. Hence, NNs will be employed to build the relationship between deformation and sensor readings.



**Fig. 3.** Sensor readings from two strain gauges under load-free and load scenarios, with the third chamber at rest.  $P_1$  and  $P_2$  (unit: bar) are the pressure of two chambers, respectively.

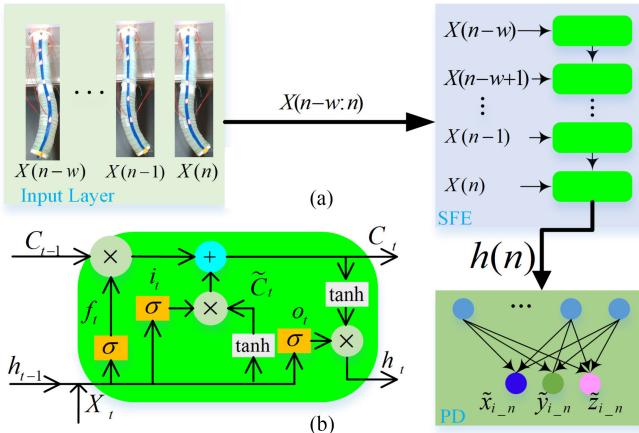


**Fig. 4.** Illustration of continuum robot being disturbed by external forces, leading to spatial backbone not a simple arch.

### III. SHAPE RECONSTRUCTION

To reconstruct the shape of continuum robot both in load free and load scenarios, a data-driven approach was considered in this work. When a continuum robot is experiencing external payloads, as shown in Fig. 4, the shape does not necessarily follow the PCC approximation (the projection in  $XOY$  is not a line). An alternative way to reflect the shape is through the backbone, and the SR problem can be converted as finding the position of some points located on the backbone, and then a curve fitting method is applied to connect these key points together.

To balance between the accuracy and the computation burden, one continuum robot with length  $L$  is divided into  $N$  subsegments, and the  $j$ th of which is labeled as  $l_j$  (see Fig. 4). The curvature of a subsegment can be represented by three key points ( $P_{2j-2}, P_{2j-1}, P_{2j}$ ). Therefore, the first step is to find the position of all key points, and the second step is to estimate the backbone using the key points with proper fitting algorithm.



**Fig. 5.** (a) Framework of SFE-PD NN (For one key point prediction). (b) LSTM cell.

### A. Key Points Prediction

Due to the uncertainty of external loads and imperfection of fabrication, it is challenging to find the position of the key points explicitly. As the actuator inputs and sensor readings are both related to the shape of robot, a data-driven approach was employed to evaluate the position of the key points and factors involved in them. Actuation inputs and external payloads (reflected via sensor readings) were both regarded as the factors. Let \$A\_n\$ and \$S\_n\$, respectively, denote the actuator inputs and the sensor readings in step \$n\$, and the true position of the \$i\$th key point \$P\_{i\\_n}\$ is \$(x\_{i\\_n}, y\_{i\\_n}, z\_{i\\_n})\$. Let \$X(n) = \{A\_n, S\_n\}\$ denote the input and \$Y(n) = P\_{i\\_n}\$ is the data to be estimated. Since the positions of different key points are sequentially dependent to each other over time, the temporal relationship (\$w\$ steps before \$n\$) between data are included, and the signal data and output data for the key point \$P\_{i\\_n}\$ in step \$n\$ are

$$\begin{cases} X(n-w:n) = \{A_{n-w}, S_{n-w}, \dots, A_n, S_n\} \\ Y(n) = \{x_{i\_n}, y_{i\_n}, z_{i\_n}\} \end{cases}. \quad (1)$$

We set \$w = 5\$ in this work. A separate NN is used to train and evaluate each key point, as shown in Fig. 5(a). Instead of setting up a single NN to predict all key points together, splitting the large data to train their respective NN can help to reduce the dimension of the model, as well as the possibility of overfitting. The network architecture mainly includes two components: 1) Sequence feature extractor (SFE) and 2) position decoder (PD). First, the SFE extracts all the input data and obtains a feature vector \$h(n)\$, which represents the status at current step and previous information. Then, PD decodes this feature and estimates the position of the key point.

*1) SFE Layer:* The SFE Layer is constructed using long short-term memory neural networks (LSTM-NNs), which is powerful in processing sequence data. A basic illustration of LSTM cell is shown in Fig. 5(b). There are three gates, namely input gate, forget gate and output gate, and they work effectively to memorize pheromone in sequence and selectively forget unimportant features. It can be mathematically expressed as

follows:

$$\begin{aligned} f_t &= \sigma(W_f X_t + w_f h_{t-1} + b_f), i_t = \sigma(W_i X_t + w_i h_{t-1} + b_i) \\ \tilde{C}_t &= \tanh(W_c X_t + w_c h_{t-1} + b_c) \\ O_t &= (W_o X_t + w_o h_{t-1} + b_o) \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \\ h_t &= O_t \cdot \tanh(C_t) \end{aligned} \quad (2)$$

where \$W\_f, W\_i, W\_c, W\_o\$, respectively, denote the weight matrices of \$X\_t\$, and \$w\_f, w\_i, w\_c, w\_o\$ are the weight matrices of \$h\_{t-1}\$. \$b\_f, b\_i, b\_c, b\_o\$ are bias vectors, and \$\sigma(\cdot)\$ is sigmoid function. The output of this SFE layer is \$h(n) \in R^{6 \times 1}\$, which is the result of the last LSTM cell.

*2) PD Layer:* This component is a fully connected NN, which has a hidden layer with six nodes and an output layer with three nodes. In the hidden layer, the \$k\$th node processes the sequence feature \$h(n)\$ by

$$f_k(h(n)) = \sigma(u_k \cdot h(n) + b_k) \quad (3)$$

where sigmoid is activation function, and \$u\_k, b\_k\$ are weight and bias. In the output layer, there are three nodes representing the position of the predicted key point \$\tilde{P}\_i(\tilde{x}\_{i\\_n}, \tilde{y}\_{i\\_n}, \tilde{z}\_{i\\_n})\$, each of which is obtained using linear activation function. It is calculated by

$$\tilde{T}_{i\_n} = \sum_{k=1}^6 u_{jk} f_k + b_{jk} \quad (j = 1, 2, 3) \quad (4)$$

where \$\tilde{T}\_{i\\_n}\$ denotes an item in the output layer, and \$u\_{jk}, b\_{jk}\$ are weight and bias of the \$j\$th node to the \$k\$th node in the hidder layer, respectively.

During the implementation, to avoid overfitting, a dropout layer with rate of 0.3 is designed between SFE and PD. The Adam optimization algorithm was used in training and the initial learning rate was set to 0.001. RMSE was used to define the loss function of the \$i\$th SPD-PD NN in training, which is

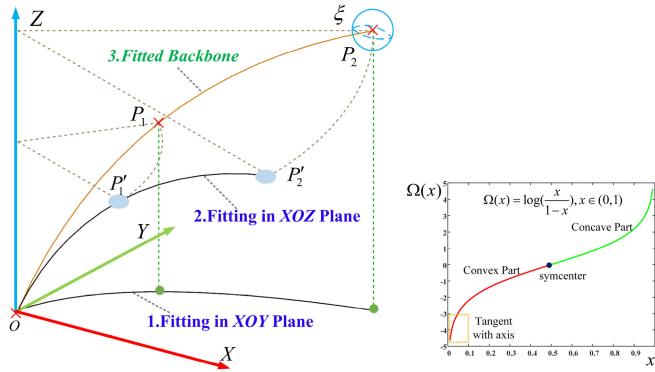
$$L_i = \sqrt{\sum_{j=1}^M \frac{1}{M} (\tilde{x}_{i\_j} - x_{i\_j})^2 + (\tilde{y}_{i\_j} - y_{i\_j})^2 + (\tilde{z}_{i\_j} - z_{i\_j})^2} \quad (5)$$

where \$M\$ is the total number of key points in training, and \$\tilde{P}\_i(\tilde{x}\_i, \tilde{y}\_i, \tilde{z}\_i)\$ denotes the predicted position. Many existing works did not consider the relationship between fitting accuracy and the accuracy of the key points [44]. Since the error between \$P\_i\$ and \$\tilde{P}\_i\$ always exists, the maximum position error \$\xi\$ in all key points

$$\xi = \max_{1 \leq i \leq 2N} \|\tilde{P}_i - P_i\| \quad (6)$$

where there are \$2N\$ key points in each fitting, and the error can be illustrated in Fig. 6(a) (blue sphere). Our two-segment continuum robot was divided into three subsegments, with six key points to be predicted (the origin \$O\$ is assumed fixed, and six SFE-PD NNs are required).

### B. Backbone Fitting



**Fig. 6.** (a) Illustration of the first subsegment, whose backbone is projected onto  $XOY$  and  $XOZ$  planes for fitting separately. (b) Basic illustration of ISF. This function is smooth and the initial part (marked by yellow rectangle) is tangent with  $Y$  axis, so it is applied to fit  $Z_j(x, y)$ .

Fig. 6(a) shows the backbone of the first subsegment, in which the three key points are  $O, P_1, P_2$ . For any subsegment  $l_j$ , it can be represented by a spatial function  $z_j = \psi_j(x, y)$ . A two-stage evaluation scheme was considered and the backbone was first projected onto the  $XOY$  plane and the  $XOZ$  plane to evaluate the  $y$  coordinates and  $z$  coordinates in terms of  $x$ . The new spatial curve functions become

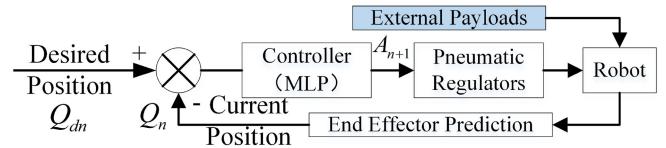
$$\begin{cases} y = Y_j(x) \\ z = Z_j(x, y) \end{cases}. \quad (7)$$

1) *Fitting in  $XOY$  Plane:* Since the backbone projection on the  $XOY$  plane should exhibit a curvilinear shape, a parabolic equation was considered to approximate the  $Y_j(x)$  function. For the key points with  $x$ -coordinates ( $x \in [x_{2j-2}, x_{2j}]$ ), the function can be expressed as:  $Y_j(x) = A_j x^2 + B_j x + C_j$ , where  $A_j, B_j, C_j$  are parameters to be solved. This function should satisfy two constraints: 1) output of this function at the three key points must be equal to their respective true  $y$  value, and 2) the slope at the starting part of the segment should be equal to the slope at the end of the previous segment  $Y_{j-1}(x)$  for continuity. With four constraints and three parameters, an optimization algorithm was used to solve for the unknown parameters, which is

$$\begin{aligned} & \text{minimize } \sum_{i=2j-1}^{2j} \|Y_j(x_i) - y_i\| \\ & \text{s.t. } \begin{cases} Y_j(x_{2j-2}) = y_{2j-2} \\ Y'_j(x_{2j-2}) = Y'_{j-1}(x_{2j-2}) \end{cases} \end{aligned} \quad (8)$$

2) *Fitting in  $XOZ$  Plane:* The three key points were then projected onto the  $XOZ$  plane and  $P_i(x_i, y_i, z_i)$  was updated to:  $P'_i = (\sqrt{x_i^2 + y_i^2}, 0, z_i)$ , as depicted in Fig. 6(a). Due to the large variation on the  $z$ -curve for different backbone shapes, the inverse of sigmoid function (ISF) was considered to approximate the  $z$  function. The basic illustration of ISF is illustrated in Fig. 6(b), and the fitting can be expressed as

$$\begin{cases} t = \sqrt{x^2 + y^2} \\ Z_j(x, y) = Z_j(t) = D_j \log \left[ \frac{t - E_j}{F_j - (t - E_j)} \right] + G_j \end{cases} \quad (9)$$



**Fig. 7.** Control scheme with feedback mechanism.

where  $x \in [x_{2j-2}, x_{2j}]$ ,  $y \in [y_{2j-2}, y_{2j}]$ , and  $D_j \dots G_j$  are parameters to be solved. Similar to fitting  $Y_j(x)$ , a constrained optimization algorithm was applied to solve the unknown parameters, which is

$$\begin{aligned} & \text{minimize } \sum_{i=2j-1}^{2j} \|Z_j(x_i, y_i) - z_i\| \\ & \text{s.t. } \begin{cases} 0 < D_j \\ E_j < \min(t) \\ 2 \max(t - E_j) < F_i \\ Z_j(x_{2j-2}, y_{2j-2}) = z_{2j-2} \\ Z'_{j-1}(x_{2j-2}, y_{2j-2}) = Z'_j(x_{2j-2}, y_{2j-2}) \end{cases} . \end{aligned} \quad (10)$$

The five constraints are summarized as follows: 1) to ensure that  $Z_j(x, y)$  is positive; 2) to ensure the numerator of (9) is positive; 3) to ensure  $Z_j(x, y)$  is in the convex part of ISF, which is similar to the backbone; 4) to ensure the continuity between  $Z_j(x, y)$  and  $Z_{j-1}(x, y)$ ; and 5) to ensure the continuity of slope. Note that the last constraint in (10) is only feasible for  $j \geq 2$ , and  $Z'_1(x_0, y_0) = \tan(89^\circ)$  was set for  $j = 1$ .

#### IV. CONTROL SCHEME

The properties of a flexible manipulator under uncertain external forces make it difficult to control the robot accurately. Hence, closed-loop control was incorporated to deal with this problem. Based on the position estimated from the SR module, a control scheme was developed as shown in Fig. 7. It should be noted that robot's tip is the key point  $P_{2N}$  predicted by one *SFE-PD* NN, such that real-time tip position can be used for feedback in control.  $Q_n$  and  $A_n$ , respectively, denote the current tip position and actuation inputs in step  $n$ , and  $A_{n+1}$  refers to the theoretical actuation inputs computed by the controller for a given desired tip position  $Q_{dn}$ .

##### A. Jacobian-Based Control Scheme

In robotics, the Jacobian matrix can be used to map between the velocity of end effector and the actuator configuration

$$Q'_n = JA'_n \quad (11)$$

where  $J$  is Jacobian matrix, and  $(\cdot)'$  is the first derivative. Theoretically, within a short time interval, (11) can be rewritten as

$$Q_{n+1} - Q_n \approx J(A_{n+1} - A_n). \quad (12)$$

Using (12), the required  $A_{n+1}$  for a given  $Q_{dn}$  can be computed as:  $A_{n+1} \approx J^{-1}(Q_{dn} - Q_n) + A_n$ , where  $Q_{dn}$  is close to  $Q_n$ . However, Jacobian matrix varies at different locations and uncertain external disturbances deteriorate the accuracy of  $J$ . The

term  $J$  can be updated to  $J_n$  to represent the unique value at step  $n$ , and  $A_{n+1}$  can be obtained via

$$A_{n+1} = J_n^{-1}(Q_{dn} - Q_n) + A_n. \quad (13)$$

The challenge for this solution is to precisely find  $J_n$  at different locations, though external payloads have not been considered. To address this issue, a nonlinear mapping function  $g_n(\cdot)$  was designed taking sensor reading  $S_n$  into account, as  $S_n$  is related to the current tip position  $Q_n$ , so that

$$A_{n+1} = g_n(Q_{dn}, Q_n, A_n, S_n) \quad (14)$$

where  $\|Q_{dn} - Q_n\| = d$  is step distance that should be smaller than an user-defined value, such that  $g_n(\cdot)$  only holds within a small area. A learning approach was implemented to find the required actuator inputs based on the known information, and  $g_n(\cdot)$  can be replaced by a well trained NN framework to cover the whole task space. Notably, in work [45], the mapping relationship is assumed to be locally linear, which is proven to be robust to against disturbances. Therefore, the local mapping which also factors the loading effect can be updated to

$$A_{n+1} \leftarrow g(Q_{dn}, Q_n, A_n, S_n). \quad (15)$$

Since this local mapping is only feasible within a small region, for a given destination  $Q_d$  located far from  $Q_n$ , robot system should plan the temporal position  $Q_{dn}$  via

$$Q_{dn} = Q_n + d \frac{\overrightarrow{Q_n Q_d}}{\|\overrightarrow{Q_n Q_d}\|} \quad (16)$$

and the end-effector will gradually move to the desired destination  $Q_d$ . The robot can stop moving if  $\|Q_d - Q_n\| \leq E$ , in which  $E$  is the minimal acceptable error between current position and destination. To adapt dynamic environment better, a dynamic step distance  $d$  can be considered based on the external payloads and to minimize the error between  $Q_{dn}$  (the planned position in step  $n$ ) and  $Q_{n+1}$  (the true position in step  $n+1$ ). Consider a larger  $d$  when external payloads influence seriously, while use a smaller  $d$  if approaching the destination.

### B. Adaptive Step Distance Mechanism (ASDM)

To achieve the dynamic step distance, the step distance  $d$  can be replaced to  $d_n$ , which can be adjusted according to the error  $e_n$

$$e_n = \|Q_{d(n-1)} - Q_n\|. \quad (17)$$

This error reflects the distance between the planned position  $Q_{d(n-1)}$  in step  $n-1$  and the true position  $Q_n$  in step  $n$ , which further shows the influence of external disturbance. The error was considered to update the  $d_n$  adaptively, and the equation is

$$d_n = d_{n-1} + \text{sgn}(\overrightarrow{Q_{n-1} Q_n} \cdot \overrightarrow{Q_{n-1} Q_{d(n-1)}}) \cdot K \cdot e_n \quad (18)$$

where  $\text{sgn}(\cdot)$  is sign function and  $K$  is a gain constant regulating the influence from position error  $e_n$ . To avoid the instability of the local controller, we limit  $2 \text{ mm} \leq d_n \leq 15 \text{ mm}$ , and  $K$  can be obtained empirically.

**TABLE I**  
MATERIAL PROPERTIES OF THE MANIPULATOR

Part	Elasticity Modulus	Poisson's Ratio	Density( $g/cm^3$ )
Soft Manipulator	15 Psi	0.48	1.07
Ring and Fixator	3 GPa	0.35	1.29

### C. Architecture of Locally Learned Controller

A multilayer perceptron (MLP) was used to build the mapping  $g(\cdot)$ . The input layer has 18 nodes ( $A^{6 \times 1}, Q_n^{3 \times 1}, Q_{dn}^{3 \times 1}, S_n^{6 \times 1}$ ) and the output layer has six nodes for  $A_{n+1}^{6 \times 1}$ . Two hidden layers are attached after the input layer with size of 7 and 5, respectively. The activation function of hidden layer is also sigmoid. Rectified linear unit (ReLU) is selected as the activation function of the output layer because the output pressure of the pneumatic regulators is always nonnegative, and ReLU is:

$$\text{ReLU}(x) = \max(0, x). \quad (19)$$

Thus, the loss function denoting the true Actuation inputs  $A$  and the predicted actuation inputs  $\tilde{A}$  in training can be derived as

$$L(A, \tilde{A}) = \sqrt{\frac{1}{6M} \sum_M \sum_{k=1}^6 (A(k) - \tilde{A}(k))^2} \quad (20)$$

where  $M$  is the number of training data and  $k$  is the  $k$ th pneumatic regulator.

## V. SIMULATION

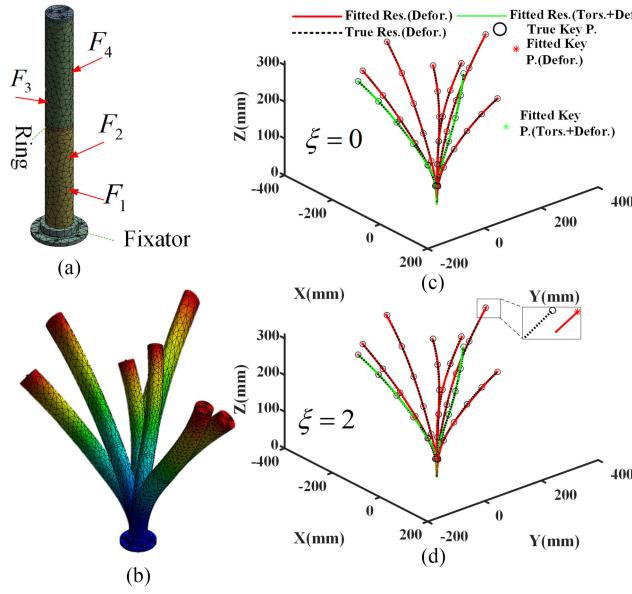
To examine the performance of the proposed SR method, it was first compared with the shape generated by finite element method (FEM). The control scheme was also compared with the performance based on piecewise constant curvature model.

### A. SR Simulation Using FEM

The error function used to evaluate the position deviation on the shape between the fitted backbone and the backbone from FEM can be defined as

$$Err = \frac{1}{60} \sum_{k=1}^{60} \|P_{true\_k} - P_{fitted\_k}\| \quad (21)$$

where 60 points on the backbone are used to compare the shape and their values can be obtained from the FEM simulation, denoted as the true (theoretical) backbone,  $P_{true\_k}$ , and through the fitting algorithm, denoted as  $P_{fitted\_k}$  accordingly. The material properties used in the simulation are listed in Table I. Other parameters and settings are learned from literature [46]. Ogden material model was selected to model the stress-strain behavior of the soft body, and the body was meshed using tetrahedral elements, as depicted in Fig. 8(a). A total of 20254 linear elements and 38212 nodes were generated. The centerline along the backbone was extracted to evaluate the positions of the true backbone (including 60 points), as well as the seven key points (gapped with 50 mm) used in the fitting algorithms. Six chambers were used to actuate the two segments, where numbers 1–3 are supplied to the proximal chambers, and numbers 4–6 are supplied to the distal chambers. Four forces (1N) that were



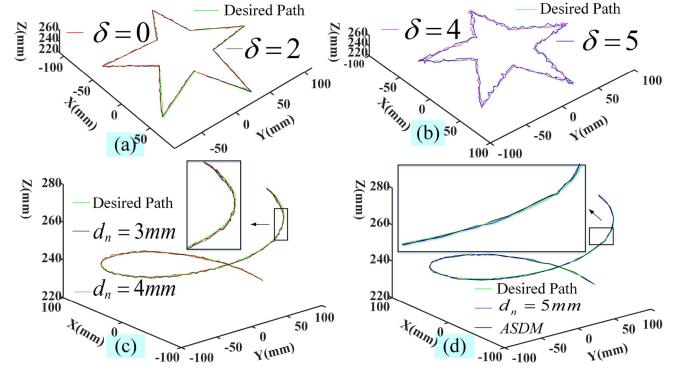
**Fig. 8.** Illustration of FEA results. (a) Meshed plot of continuum robot, with four external forces. (b) Deformation of the robot under six scenarios. (c) and (d) Comparison of fitted backbone and true backbone.

**TABLE II**  
MEAN ERROR OF EACH FITTED BACKBONE WITH DIFFERENT ERROR OF KEY POINTS (UNIT:mm)

Actuated Chambers	$\xi = 0$	$\xi = 2$	$\xi = 3$	$\xi = 4$
1,4	2.83	3.08	3.29	3.46
1,5	3.16	3.88	3.97	3.61
2,4	2.91	3.31	3.58	3.68
2,5	3.18	3.23	3.35	3.59
3,5	2.55	2.69	2.92	3.34
3,6	2.69	2.73	3.17	3.41
1,2,4,5(Tors.)	2.42	2.76	3.21	3.53
2,3,4,6(Tors.)	2.53	2.87	3.32	3.59

evenly distributed along the 300 mm long backbone (60 mm apart), and they were 90° shift to spirally act on the manipulator, mimicking the external payloads [see Fig. 8(a)]. Eight different actuation configurations covering the entire task space were examined, where each actuated chamber is pressurized with 1 bar at a time.

To mimic error in estimating the six key points through NNs, a random error  $\xi$ , in a range between 0 to 4 mm, was deliberately added to the six key points (except the origin), and new key points used in the fitting algorithm become  $\tilde{P}_i = P_i + U(-\xi, \xi)$ , where  $U(\cdot)$  is a uniform distribution. The results of the bending status are shown in Fig. 8(c) and (d). The acting position of  $F_1$  is changed to the edge of the manipulator, such that torsion and deformation occur simultaneously. It can be observed that the shape from the fitting algorithm is matched with the FEM simulation. The fitting error under different actuation configurations are listed in Table II. It can be seen that when the error of the key points increases (i.e., a larger  $\xi$ ), the fitting curve starts to deviate from the FEM simulation and a larger  $Err$  is observed, as summarized in Table II. Based on the results, the maximum error of the estimated key points should be limited within 3 mm



**Fig. 9.** Simulation results for the control scheme. (a) and (b) the path following tests on a star-shaped path, where noise is variable. (c) and (d) tests on a spiral path, where noise is constant and different step distance is used.

or the estimated shape from the proposed algorithm will not be accurate enough.

### B. Path Following Simulation

To test the proposed control scheme in a simulation manner, a computer model for a two-segment continuum robot based on piecewise constant curvature assumption was set up for collecting training data and verifying the performance of the trained model. During data collection, the length of each chamber is commanded to vary within [140 mm, 160 mm]. It should be noted that all the six chambers can only elongate by 2 mm or -2 mm randomly in an exploration, to ensure that adjacent position data are located nearly. As a result, 4000 training data are obtained in load-free manner. Since sensor readings are not available in the simulation, the input layer of MLP has reduced to only 12 nodes ( $A_n \in R^{6 \times 1}$ , and  $Q_n, Q_{dn} \in R^{3 \times 1}$ ). Other settings about training controller is the same as that defined in Section IV. Two experiments related to testing disturbance handle ability and the performance of ASDM were conducted, where  $E$  is 2 mm and  $K$  is 0.12 (through empirical setting).

First, a star-shaped path following task was prepared, including continuous discrete points on the path, and end effector is commanded to arrive them one by one. The step distance  $d_n$  is set as 5 mm, and different disturbance is considered for comparison. The effect from external payload was simplified as error leading to the position offset on the end-effector, which means  $Q_n \leftarrow Q_n + N(0, \delta)$  ( $N(0, \delta)$  is Gaussian white noise with mean of zero and variance of  $\delta$ ). The results are shown in Fig. 9(a) and (b). When  $\delta = 0$ , the true path is almost consistent with the desired path, and when a larger  $\delta$  was considered, the actual path gradually deviates the desired path, but end effector can still reach destinations and complete the path.

Second, the feasibility of the proposed ASDM was examined, where the noise  $\delta$  is set to 5, and the step distance  $d_n$  was updated adaptively. In this test, a spiral path is prepared. We initially set  $d_n = 4$  mm to follow the desired path. As shown in Fig. 9(c), the end effector fluctuates near  $Q_{dn}$ , and it takes 182 steps to finally arrive the destination. Several phenomenon can be observed from the simulation. First, when a smaller step

distance ( $d_n = 3$  mm) was selected, the effect from the external payload on the end-effector becomes more obvious, leading to a larger error. When the step distance increases to 5 mm, and it only took 125 steps and the actual path is closer to the desired path, as shown in Fig. 9(d). When the ASDM was adopted, the robot only needed 116 steps to complete the path. The fluctuation on the end-effector is very minimal, showing the robustness of the proposed adaptive controller to handle external disturbances.

## VI. EXPERIMENTAL RESULTS

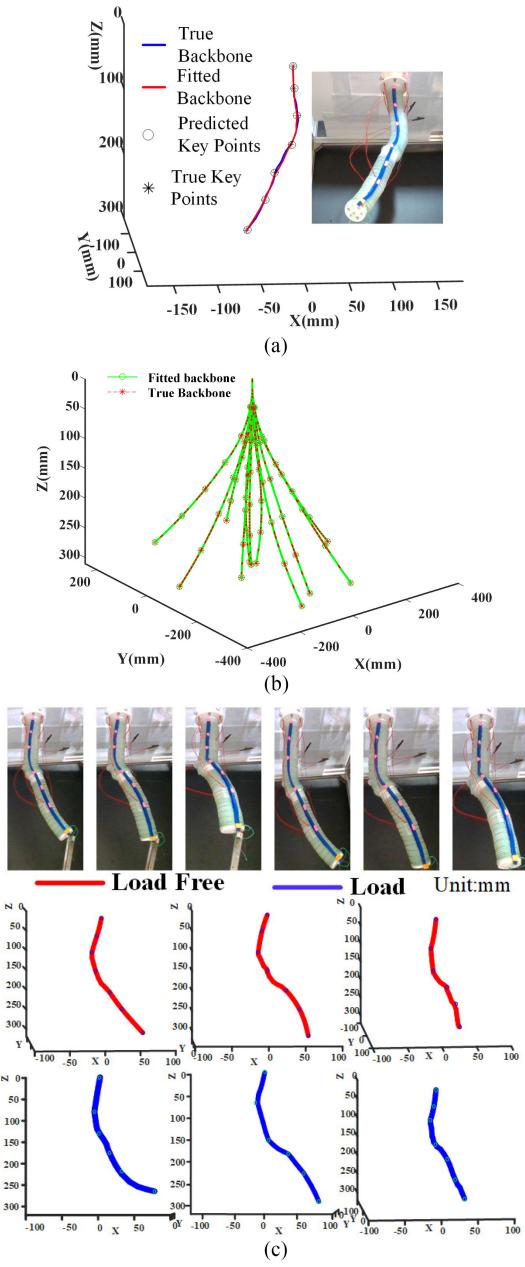
While the results from simulation confirm the feasibility of the proposed methodology, the algorithms were implemented onto the robot system to examine experimentally. Since both the SR module and control scheme are built using data-driven approach, training data was sampled first, and the actual performance of the two modules are, respectively, evaluated. In testing, the RGB-D camera only collected true position of backbone and key points for comparison, providing no information for robot system.

### A. Model Preparation

The robot was commanded to randomly explore within the task space by supplying different pressures to the six chambers. The maximum pressure was limited to 0.8 bar to avoid over-pressure. At each instance, any two of the six chambers were randomly adjusted to increase or decrease the pressure by 0.05 bar so as to ensure a small, incremental change in the shape between the previous and the new state. After 1 s, the position of the backbone was obtained through an RGB-D camera, which was assumed as the ground truth, and the sensor readings were simultaneously recorded too. 6000 groups of data were collected, including the position of the backbone, the position of key points, sensor readings and actuation configurations. 4000 groups of them were collected from the free state (no load), and the remaining 2000 were from the load state. A weight with 75 g was attached to the tip of robot, resulting a 0.75 N downward pulling force at the end-effector. After data collection, each data is normalized within  $[-1, 1]$ , and the output of each NN is converted back to its original scale after inference to equalize the range of each variable. In training, each NN is commanded to be trained at most 1000 iterations or RMSE is smaller than 0.001, where the former 5500 sets were used for training and the rest was set for validation. As a result, the RMSE of SFE-PD and MLP are, respectively, 0.00093 and 0.00089, indicating the models were well trained.

### B. Characterization of SR Module

Fig. 10(a) shows the true bending state and the fitted backbone of this robot, where the six pneumatic pressure regulators are set to: 0, 0.5, 0.6, 0.8, 0.7, 0 (unit: bar). The coordinates of the corresponding key points that are sampled by RGB-D camera and predicted through SFE-PD, are listed in Table III, which confirms that the error from each key point is less than 3 mm, namely  $\xi = \|\tilde{P}_i - P_i\| < 3$  mm. The fitted backbone is almost consistent with the true backbone (obtained by RGB-D



**Fig. 10.** (a) Illustration of backbone fitting with one actuation configuration. (b) SR in the entire work space. (c) Fitted backbone in load and load-free conditions using the same actuation configurations.

**TABLE III**  
COMPARISON ON KEY POINTS BETWEEN THE TRUTH AND THE FITTED  
VALUE (UNIT:mm)

Key Point	True Position	Predicted Position	Error
$P_0$	(0,0,0)	Needn't Prediction	
$P_1$	(-0.1,0.5,49.6)	(-0.1,0.4,48.3)	1.31
$P_2$	(4.2,4.9,90.1)	(3.9,4.35,91.2)	1.3
$P_3$	(3.8,9.6,133.2)	(4.1,10.3,135.2)	2.14
$P_4$	(-22.6,19.9,172.9)	(-21.7,21.1,172.1)	1.7
$P_5$	(-35.9,35.2,210.8)	(-36.2,34.1,211.7)	1.45
$P_6$	(-58.7,64.7,248.5)	(-59.2,65.3,250.2)	1.51

**TABLE IV**

FITTING RESULTS BETWEEN LOAD FREE AND LOAD CONDITIONS (MAXIMUM ERROR/MEAN ERROR, UNIT: mm)

weight	1	2	3	4
free load	6.87/5.63	6.67/5.78	6.92/6.01	6.58/5.57
50g	6.78/5.64	6.71/5.81	6.94/5.97	6.73/5.62
75g	7.06/6.13	6.94/6.03	7.11/6.21	6.93/6.34
119g	7.12/6.24	7.26/6.31	7.34/6.41	7.05/6.28
150g	7.32/6.56	7.38/6.49	8.03/6.94	7.87/6.45

camera). The fitting task needs only 0.28s in MATLAB 2019b Optimization Toolbox, which can almost ensure the SR module is working in real time. The mean error calculated by (21) is 5.85 mm. Another 11 actuation inputs were set to get more shape states covering the whole task space for comprehensive testing. Fig. 10(b) shows the fitted results and the true backbone, where the fitting errors are consistent with the previous testing. This further confirms the accuracy and feasibility of the proposed SR method.

The fitting results in load scenarios are also our focus, so we also compared the error in load and load-free conditions, where actuation inputs in one comparison are same but loads differ each other. For the load experiments, weight with 50, 75, 119, and 150g were hung at the tip, where same actuation inputs were applied in an independent scenario to compare the position errors. Fig. 10(c) shows the snapshots and the fitted backbone for four different actuation inputs, where the weight is 119 g. The mean position error between the fitted backbone and the true shape(set four different actuation inputs and each of them consists of four external payloads), were computed and summarized in Table IV. From the results, the errors from the load scenarios are generally higher than that in load-free scenarios, indicating the external load can lead to larger errors in the prediction. However, the increase in the error is small, showing the proposed method is able to predict the key points for both load and load-free scenarios.

### C. Point-to-Point Movement Task

The SR module can effectively provide accurate information on the position of the robot system, particularly the end-effector. In this experiment, the end effector was set to move from point (40, 20, 300) to point (-50, 70, 210) (unit: mm).  $E$  is set to 4mm, and  $K$  is also set to 0.12 empirically.

1) *Performance of Addressing Different Disturbances:* To investigate the performance of ASDM and the performance of the controller, four conditions were considered to achieve this point-to-point movement task, namely: I) $d_n = 8$ , II) $d_n = 2$ , III)ASDM( $d_1 = 8$ ), IV)ASDM with different loads (statically hanging a tip load with 50, 75, and 100g). Fig. 11(a) and (b) shows the true path of (I)–(II) and (III)–(IV), respectively. Fig. 11(c) shows the actual distance to destination for the four scenarios. When  $d_n = 8$ , the end-effector can reach to the desired point after 24 steps but it starts to fluctuate when approaching the destination (in the last 4 steps). In contrast, when  $d_n = 2$ , the end-effector can steadily reach the desired point but it took 75 steps. When ASDM was applied, the end-effector

moves directly to the destination in 14 steps, and Fig. 11(d) illustrates corresponding change of actuation configuration of (III). Fig. 11(i) shows the snapshots of the (III) scenario. Therefore, the results from the former three conditions prove the feasibility of ASDM. In condition (IV), when tip load was not beyond 50 g, robot followed the desired path well, and when the load increased, the true path deviated the desired more at initial steps, but it gradually moved toward estimation, which demonstrates the controller can address external payloads.

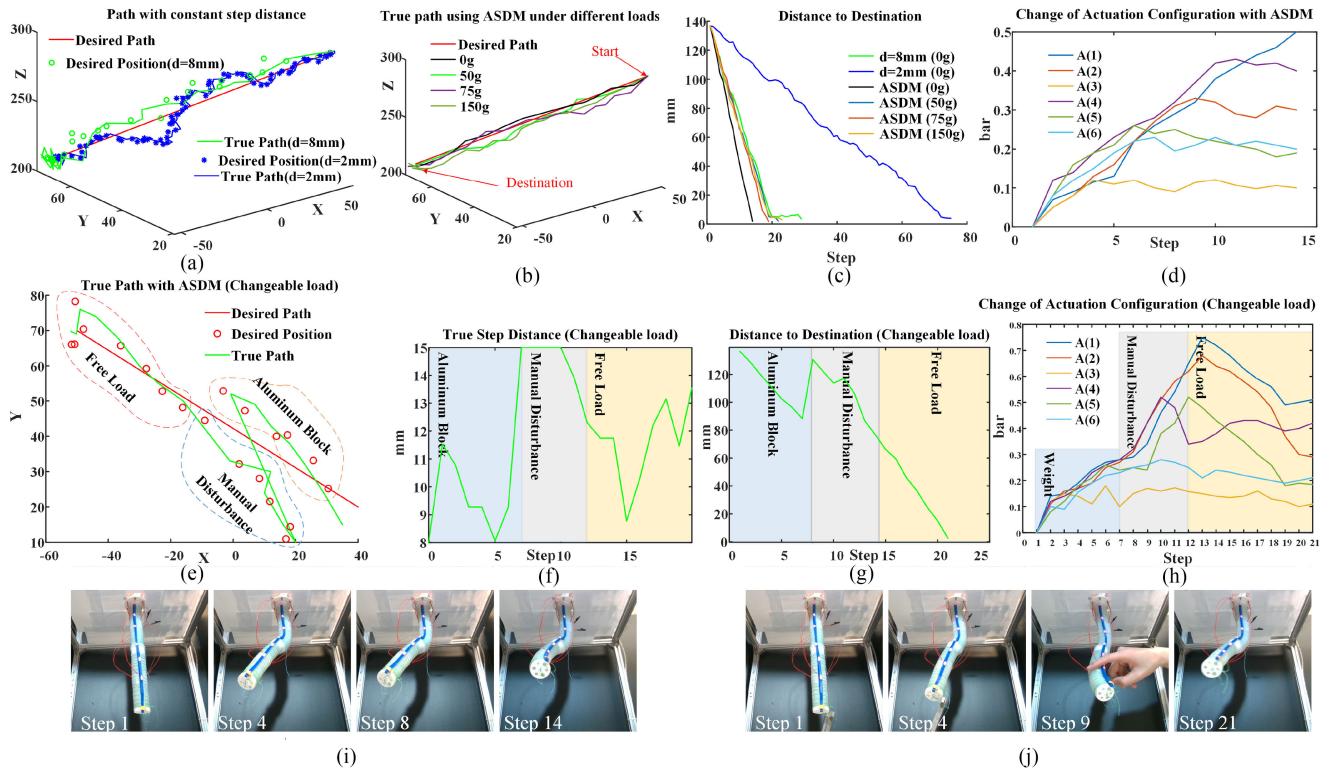
2) *Performance of Addressing Changeable Payloads:* An additional experiment was conducted to test hybrid payloads. Initially, a swaying aluminium block (119 g), was attached to the end-effector and caused position deviation from the desired path, but the error was generally reduced. At the 7th step, the aluminium block was replaced by a manual disturbance acting at the mid-point of the manipulator, which pulled the end effector to a new position (19.8, 10.2, 303.6). At this moment, robot system sensed the disturbance then increased the step distance to 15 mm [see Fig. 11(g) and (h)]. From step 7 to step 12], the robot tried to reject the external disturbance by adjusting the actuation configuration. At step 12, the manual disturbance was removed, and the robot gradually corrected the error and followed the planned path to reach the desired position. Fig. 11(e) shows the true path from top view. The actuator inputs are comparable to the load-free scenario once the load was removed at step 12 [see Fig. 11(d) and (h)]. This shows that our controller can also work when faced with changeable external payloads.

### D. Path Following Task

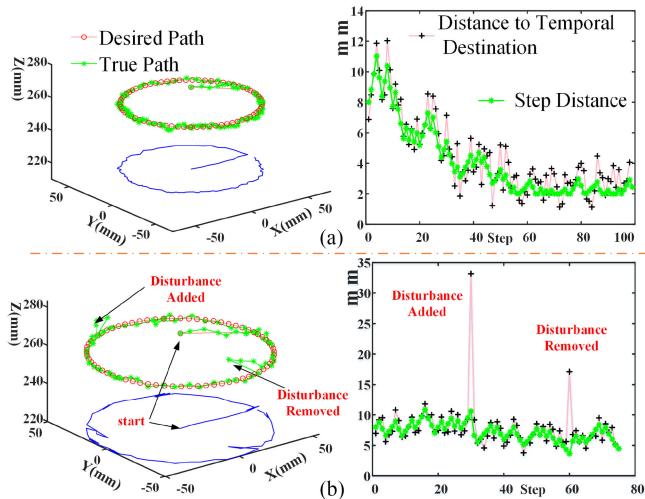
Path following tasks were also examined with the proposed controller, and the path is composed of continuous path points. A circular path with a diameter of 100 mm, was first examined. The initial step distance was set to 8 mm, as shown in Fig. 12(a). Experimental results confirm that the end-effector can follow closely to the desired path, and the step distance changes adaptively. Second, external disturbance was added during the operation along the same path, and the results are shown in Fig. 12(b). At the 29th step, an aluminium block (with 263 g) was added at the end-effector, making its position deviate sharply from the desired path, but robot system gradually adjusted the step gap distance to follow the target points. At the 60th step, the aluminium block was removed, leading to a large position error again [see Fig. 12(b)]. The robot can still handle it and return to the desired path after four steps. Therefore, the proposed ASDM plays a critical role in overcoming the influence from external payloads.

### E. Discussion

Time efficiency is a critical concern in motion. The time costs are about 0.013527 and 0.007275 s, respectively, for one SFE-PD and the MLP-based controller (Intel i7 processor with NAVIDIA GeForce MX250), such that the time cost of NNs computation is 0.0884 s. It averagely needs 0.28 s to fit robot's backbone. The total time cost is 0.37 s for one epoch. Furthermore, the shape estimation and control methods can be extended to robots with different dimensions and



**Fig. 11.** Results of point-to-point movement task. (a)–(d) load-free scenario. (e)–(h) robot is moving with uncertain external payload. (i)–(j) snapshots of load-free state and uncertain disturbance state.



**Fig. 12.** Results of path following task, including true path corresponding step distance, and distance to temporal destination. (a) Free-load state. (b) Disturbance-added state.

configurations. Since the training data is unique to a particular robot configuration, new training datasets should be collected accordingly. Different scenarios, including load-free, point-load and distributed load, can all be included in collecting the data for training.

The maximum shape sensing error in our work is around 8 mm with OD 35 mm and length 310 mm. The error is highly involved in the length and OD. In comparison, most existing

works obtained the key points directly using multiple sensors, such as RMSE=13 mm using vision system (Length: 287 mm, OD: 24 mm) in [47]. In [44], the error is 2.27 mm using FBG with length around 40 mm. The error of the key points is additionally considered in this work. As for the controller, it can adapt to different uncertain loads: point load, distributed load, and mixing load. The control error is less than 4 mm for this longer-and-bigger-diameter manipulator, which is comparable with other slender ones (2 mm in [48], 2.89% of length 160 mm [49], and 2.23% of length 880 mm [40]), so we believe this control is of high precision.

## VII. CONCLUSION

This article presents two important modules that are relevant toward precise control of a continuum robot subject to external payloads, namely the real-time SR and the local inverse kinematics model. Based on the actuator inputs, *SFE-PD* NNs were first employed to estimate the different key points along the backbone and the shape was estimated through a fitting algorithm. Then, strain gauge signals were incorporated into *SFE-PD* NNs model so as to compute the required actuator inputs to bring the end-effector to the desired position. An adaptive update scheme was used in the controller so that it can minimize the effect from external payloads. Simulation and experimental results both confirm that the SR module can effectively estimate the shape of the robot, while the local IK module can guide the end-effector to different positions, in both load and load-free scenarios.

## REFERENCES

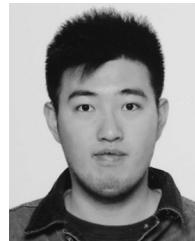
- [1] G. Robinson and J. B. C. Davies, "Continuum robots-a state of the art," *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 4, pp. 2849–2854, 1999.
- [2] C. Li, X. Gu, X. Xiao, C. M. Lim, and H. Ren, "Flexible robot with variable stiffness in transoral surgery," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 1, pp. 1–10, Feb. 2020.
- [3] X. Dong *et al.*, "Development of a slender continuum robotic system for on-wing inspection/repair of gas turbine engines," *Robot. Comput.-Integr. Manuf.*, vol. 44, pp. 218–229, 2017.
- [4] E. B. Joyee and Y. Pan, "A fully three-dimensional printed inchworm-inspired soft robot with magnetic actuation," *Soft Robot.*, vol. 6, no. 3, pp. 333–345, 2019.
- [5] Z. Wang *et al.*, "A three-fingered force feedback glove using fiber-reinforced soft bending actuators," *IEEE Trans. Ind. Electron.*, vol. 67, no. 9, pp. 7681–7690, Sep. 2020.
- [6] R. B. Schaffr, R. M. Doornbusch, E. L. Doubrovski, J. Wu, J. M. Geradts, and C. C. Wang, "Color-based proprioception of soft actuators interacting with objects," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 5, pp. 1964–1973, Oct. 2019.
- [7] C. Majidi, R. Kramer, and R. Wood, "A non-differential elastomer curvature sensor for softer-than-skin electronics," *Smart Mater. Structur.*, vol. 20, no. 10, 2011, Art. no. 105017.
- [8] T. G. Thuruthel, B. Shih, C. Laschi, and M. T. Tolley, "Soft robot perception using embedded soft sensors and recurrent neural networks," *Sci. Robot.*, vol. 4, no. 26, 2019, doi: [10.1126/scirobotics.aav1488](https://doi.org/10.1126/scirobotics.aav1488).
- [9] V. K. Venkiteswaran, J. Sikorski, and S. Misra, "Shape and contact force estimation of continuum manipulators using pseudo rigid body models," *Mech. Mach. Theory*, vol. 139, pp. 34–45, 2019.
- [10] H. Yang, M. Xu, W. Li, and S. Zhang, "Design and implementation of a soft robotic arm driven by sma coils," *IEEE Trans. Ind. Electron.*, vol. 66, no. 8, pp. 6108–6116, Aug. 2019.
- [11] J. D. Greer, T. K. Morimoto, A. M. Okamura, and E. W. Hawkes, "A soft, steerable continuum robot that grows via tip extension," *Soft Robot.*, vol. 6, no. 1, pp. 95–108, 2019.
- [12] E. Amanov, T.-D. Nguyen, and J. Burgner-Kahrs, "Tendon-driven continuum robots with extensible sections-a model-based evaluation of path-following motions," *Int. J. Robot. Res.*, 2019, Art. no. 0278364919886047.
- [13] H. Guo *et al.*, "Continuum robot shape estimation using permanent magnets and magnetic sensors," *Sensors Actuators A, Phys.*, vol. 285, pp. 519–530, 2019.
- [14] X. Ma, P. Wang, M. Ye, P. W. Y. Chiu, and Z. Li, "Shared autonomy of a flexible manipulator in constrained endoluminal surgical tasks," *IEEE Robot. Automat. Lett.*, vol. 4, no. 3, pp. 3106–3112, Jul. 2019.
- [15] N. Rahman, N. J. Deaton, J. Sheng, S. S. Cheng, and J. P. Desai, "Modular fbg bending sensor for continuum neurosurgical robot," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1424–1430, Apr. 2019.
- [16] S. Song, Z. Li, M.-H. H. Meng Yu, and H. Ren, "Real-time shape estimation for wire-driven flexible robots with multiple bending sections based on quadratic Bézier curves," *IEEE Sensors J.*, vol. 15, no. 11, pp. 6326–6334, 2015.
- [17] S. Song, Z. Li, H. Yu, and H. Ren, "Shape reconstruction for wire-driven flexible robots based on Bézier curve and electromagnetic positioning," *Mechatronics*, vol. 29, pp. 28–35, 2015.
- [18] I. Singh, Y. Amara, A. Melingui, P. Mani Pathak, and R. Merzouki, "Modeling of continuum manipulators using pythagorean hodograph curves," *Soft Robot.*, vol. 5, no. 4, pp. 425–442, 2018.
- [19] B. Ouyang, Y. Liu, H.-Y. Tam, and D. Sun, "Design of an interactive control system for a multisection continuum robot," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 5, pp. 2379–2389, Oct. 2018.
- [20] T. Li, L. Qiu, and H. Ren, "Distributed curvature sensing and shape reconstruction for soft manipulators with irregular cross sections based on parallel dual-FBG arrays," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 1, pp. 406–417, Feb. 2020.
- [21] A. Gao, N. Liu, M. Shen, M. EMK Abdelaziz, B. Temelkuran, and G.-Z. Yang, "Laser-profiled continuum robot with integrated tension sensing for simultaneous shape and tip force estimation," *Soft Robot.*, Mary Ann Liebert, Inc., vol. 7, no. 4, pp. 421–443, 2020.
- [22] H. Guo, F. Ju, and B. Chen, "Preliminary study on shape sensing for continuum robot affected by external load using piecewise fitting curves," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2019, pp. 2414–2420.
- [23] L. Scimeca, J. Hughes, P. Maiolino, and F. Iida, "Model-free soft-structure reconstruction for proprioception using tactile arrays," *IEEE Robot. Automat. Lett.*, vol. 4, no. 3, pp. 2479–2484, Jul. 2019.
- [24] R. Roy, L. Wang, and N. Simaan, "Modeling and estimation of friction, extension, and coupling effects in multisegment continuum robots," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 2, pp. 909–920, Apr. 2017.
- [25] J. Till, V. Alois, and C. Rucker, "Real-time dynamics of soft and continuum robots based on cosserat rod models," *Int. J. Robot. Res.*, vol. 38, no. 6, pp. 723–746, 2019.
- [26] K. Oliver-Butler, J. Till, and C. Rucker, "Continuum robot stiffness under external loads and prescribed tendon displacements," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 403–419, Apr. 2019.
- [27] A. Parvaresh and S. A. A. Moosavian, "Modeling and model-free fuzzy control of a continuum robotic arm," in *Proc. IEEE 6th RSI Int. Conf. Robot. Mechatronics*, 2018, pp. 501–506.
- [28] P. Qi, C. Liu, A. Ataka, H.-K. Lam, and K. Althoefer, "Kinematic control of continuum manipulators using a fuzzy-model-based approach," *IEEE Trans. Ind. Electron.*, vol. 63, no. 8, pp. 5022–5035, Aug. 2016.
- [29] 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, Aug. 2015.
- [30] 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.
- [31] G. Fang *et al.*, "Vision-based online learning kinematic control for soft robots using local gaussian process regression," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1194–1201, Apr. 2019.
- [32] B. Zhao, L. Zeng, Z. Wu, and K. Xu, "A continuum manipulator for continuously variable stiffness and its stiffness control formulation," *Mechanism Mach. Theory*, vol. 149, 2020, Art. no. 103746.
- [33] C. Wang, C. G. Frazelle, J. R. Wagner, and I. Walker, "Dynamic control of multi-section three-dimensional continuum manipulators based on virtual discrete-jointed robot models," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 2, pp. 777–788, Apr. 2021.
- [34] E. Franco and A. Garriga-Casanovas, "Energy-shaping control of soft continuum manipulators with in-plane disturbances," *Int. J. Robot. Res.*, 2020, Art. no. 0278364920907679.
- [35] 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.
- [36] G. Li, D. Song, S. Xu, L. Sun, and J. Liu, "A hybrid model and model-free position control for a reconfigurable manipulator," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 2, pp. 785–795, Apr. 2019.
- [37] M. Li, R. Kang, D. T. Branson, and J. S. Dai, "Model-free control for continuum robots based on an adaptive kalman filter," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 286–297, Feb. 2018.
- [38] C. Della Santina, R. K. Katzcsmann, A. Bicchi, and D. Rus, "Model-based dynamic feedback control of a planar soft robot: Trajectory tracking and interaction with the environment," *Int. J. Robot. Res.*, vol. 39, no. 4, pp. 490–513, 2020.
- [39] 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, 2014.
- [40] C. Yang *et al.*, "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, 2020.
- [41] J. Barrientos-Diez, X. Dong, D. Axinte, and J. Kell, "Real-time kinematics of continuum robots: Modelling and validation," *Robot. Comput.-Integr. Manuf.*, vol. 67, Art. no. 102019, 2021.
- [42] N. Tan, X. Gu, and H. Ren, "Pose characterization and analysis of soft continuum robots with modeling uncertainties based on interval arithmetic," *IEEE Trans. Automat. Sci. Eng.*, vol. 16, no. 2, pp. 570–584, Apr. 2019.
- [43] X. Gao, X. Li, C. Zhao, L. Hao, and C. Xiang, "Variable stiffness structural design of a dual-segment continuum manipulator with independent stiffness and angular position," *Robot. Comput.-Integr. Manuf.*, vol. 67, Art. no. 102000, 2021.
- [44] F. Monet *et al.*, "High-resolution optical fiber shape sensing of continuum robots: A comparative study," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 8877–8883.
- [45] J. D. Ho *et al.*, "Localized online learning-based control of a soft redundant manipulator under variable loading," *Adv. Robot.*, vol. 32, no. 21, pp. 1168–1183, 2018.
- [46] K.-H. Lee *et al.*, "Nonparametric online learning control for soft continuum robot: An enabling technique for effective endoscopic navigation," *Soft Robot.*, vol. 4, no. 4, pp. 324–337, 2017.

- [47] R. A. Manakov, D. Y. Kolpashchikov, V. V. Danilov, N. V. Laptev, I. Skirnevskiy, and O. M. Gerget, "Visual shape and position sensing algorithm for a continuum robot," *IOP Conf. Series, Materials Sci. Eng.*, vol. 1019, no. 1, 2021, Art. no. 012066.
- [48] Z. Wang *et al.*, "Hybrid adaptive control strategy for continuum surgical robot under external load," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1407–1414, Apr. 2021.
- [49] X. Huang, J. Zou, and G. Gu, "Kinematic modeling and control of variable curvature soft continuum robots," *IEEE/ASME Trans. Mechatronics*, 2021.



**Qingxiang Zhao** (Graduate Student Member, IEEE) received the B.Eng. and M.S. degrees in mechanical engineering from Sichuan University, Chengdu, China, in 2016 and 2019, respectively. He is currently working toward the Ph.D. degree in mechanical engineering with The Hong Kong Polytechnic University, Hong Kong.

His research interests include soft robotics, industrial automation, and artificial intelligence.



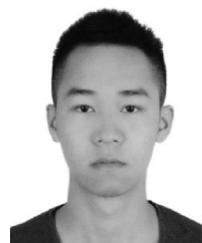
**Kaicheng Huang** (Student Member, IEEE) received the B.Eng. degree in automation from Shenzhen University, Shenzhen, China, in 2014, and the M.Sc. degree in mechanical and automation engineering from The Chinese University of Hong Kong, Hong Kong, in 2015. He is currently working toward the Ph.D. degree with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong.

His research interests include automated cell patterning with dielectrophoresis and robotics.



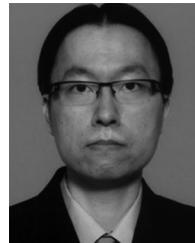
**Xiaobing Hu** received the Ph.D. degree in Mechanical Engineering from the School of Mechanical Engineering, Sichuan University, Chengdu, China, in 1999.

Currently, he is working as a professor and Ph.D. supervisor in this department. His research focuses on intelligent manufacturing, enterprise informatization, robotics, CAD/CAE/CAM, and machine vision. More than 50 M.S.C. students have been supervised by him. He was also responsible for multiple national and provincial research projects.



**Jiewen Lai** (Graduate Student Member, IEEE) received the B.Eng. degree in metallurgical engineering from the Wuhan University of Science and Technology, Wuhan, China, in 2016, and the M.Sc. degree in mechanical and automation engineering from The Chinese University of Hong Kong, Hong Kong, in 2017. He is currently working toward the Ph.D. degree in mechanical engineering with The Hong Kong Polytechnic University, Hong Kong.

His research interests include soft/continuum robot and surgical robot system.



**Henry K. Chu** (Member, IEEE) received his Bachelors degree in Mechanical Engineering (Mechatronics option) from the University of Waterloo, Canada, in 2005, and his MSc. and Ph.D. degrees in Mechanical and Industrial Engineering from the University of Toronto, Canada, in 2007 and 2011, respectively.

He was a Post-Doctoral Fellow with the University of Toronto and the City University of Hong Kong, Hong Kong. He is currently an Assistant Professor with the department of mechanical engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China. His research interests include robotic manipulation, vision-based control and automation micro-system design, and tissue engineering.