

A Two-Level Approach for Solving the Inverse Kinematics of an Extensible Soft Arm Considering Viscoelastic Behavior

Hao Jiang¹, Zhanchi Wang², Xinghua Liu², Xiaotong Chen¹, Yusong Jin³, Xuanke You¹, and Xiaoping Chen¹

Abstract—Soft compliant materials and novel actuation mechanisms ensure flexible motions and high adaptability for soft robots, but also increase the difficulty and complexity of constructing control systems. In this work, we provide an efficient control algorithm for a multi-segment extensible soft arm in 2D plane. The algorithm separate the inverse kinematics into two levels. The first level employs gradient descent to select optimized arm’s pose (from task space to configuration space) according to designed cost functions. With consideration of viscoelasticity, the second level utilizes neural networks to figure out the pressures from each segment’s pose (from configuration space to actuation space). In experiments with a physical prototype, the control accuracy and effectiveness are validated, where the control algorithm is further improved by an optional feedback strategy.

I. INTRODUCTION

Soft robots can serve as better solutions in natural and human-centric environments where safety and adaptability to uncertainty are fundamental requirements, compared with traditional rigid industrial robots, as reviewed by Rus and Tolley [1]. Utilizing their continuous deformation and theoretically infinite degrees of freedom [2], they can adaptively manipulate and grasp unknown objects varying in size and shape [3], or squeeze through confined space [4]. Nevertheless, their compliant materials, unconstrained structures and highly nonlinearity in movements increase the complexity of modeling and control.

For solving the inverse kinematics of soft robots, there are two main types of approaches: model-based and learning-based approaches. Regarding model-based methods, they mathematically formulate the system’s behavior using kinematics and dynamics equations [5][6]. Godage et al. [7] present a 3D kinematic model for multi-segment continuum arms using a novel shape function-based approach [8], which conforms geometrically constrained structure of the arm. Escande et al. [9] develop the forward kinematic model for compact bionic handling assistant (CBHA) based on the constant curvature assumption. However, many soft robot

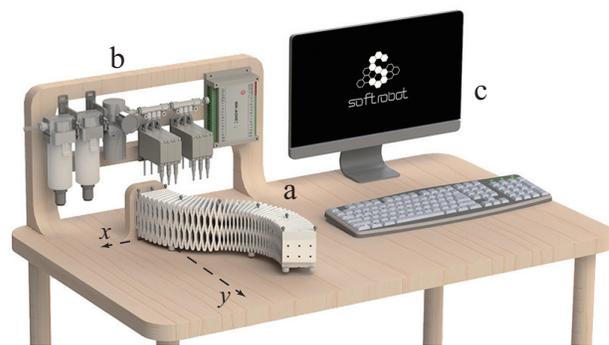


Fig. 1. An overview of the major components of the pneumatic control system: the soft extensible HPN arm (a), the pneumatic drive system (b), and the computational device (c).

models are difficult to build and less accurate mostly due to their highly nonlinear and complex structures.

Learning-based methods are widely applied in control of soft manipulators owing to their universal approximation property. Giorelli et al. [10][11] develop a feed-forward neural network to solve the inverse kinematics for a cable-driven manipulator. Braganza et al. [12] develop dynamic control based on the length sensor utilizing neural network to compensate for dynamic uncertainties on OctArm VI. Rolf et al. [13] introduce a goal babbling approach to learn an inverse kinematics of a bionic handling assistant (BHA). Generally, learning-based methods can figure out relatively accurate models for complex structures, however, for the manipulators with many DoFs and complicated structures, a large amount of training data and a long training process is essential.

In fact, a better control scheme is to combine the two previous approaches in order to take the best of them. In this work, we develop a two-level approach for inverse kinematics combining model-based method and learning-based method. From task space to configuration space, we use a model-based method to set appropriate cost function and use gradient descent algorithm to figure out optimal solutions. From configuration space to actuation space, each segment is considered individually, where a neural network learns the relationship after trained with proper amount of data. Moreover, control precision and stability are improved by taking viscoelasticity into consideration. Compared with existing neural-network based control approaches [10][11], this feature makes our approach unique. Regarding viscoelas-

*This research is supported by the National Natural Science Foundation of China under grant 61573333 and the Research Fund for the Doctoral Program of Higher Education of China under grant 20133402110026.

¹Hao Jiang, Xiaotong Chen, Xuanke You and Xiaoping Chen are with the School of Computer Science, University of Science and Technology of China, Hefei, 230026, China. jhjh@mail.ustc.edu.cn

²Xinghua Liu and Zhanchi Wang are with the School of Nuclear Science, University of Science and Technology of China, Hefei, 230026, China.

³Yusong Jin is with the School of Physics, University of Science and Technology of China, Hefei, 230026, China.

tic behavior, it is a hysteresis effect of actuators and memory phenomenon of the whole structure, impairing the control precision of most manipulators made of soft materials. In the last, a simple feedback strategy is proposed to turn the open-loop control algorithm into a closed-loop one, which further improves the control precision. In summary, the contributions of this paper include:

- A two-level approach for open-loop control of a multi-segment extensible soft arm.
- Pose optimization for the soft arm based on designed cost function.
- Neural-network based algorithm figuring out the pressures from each segment's pose with consideration of viscoelastic behavior.
- A simple strategy turning the open-loop control method into a closed-loop one with high accuracy.

II. SOFT ARM

To start, we provide the readers with a brief overview of the soft arm (Fig. 1a) developed by the authors in [14], whose conceptual structure called honeycomb pneumatic network (HPN) is proposed in [15][16]. As shown in Fig. 2, the soft and extensible arm consists of a deformable honeycomb-shaped frame (a) and airbags (b). The whole arm is composed of several segments with different physical parameters. Reflective markers (c) are located at the interface between the segments. Throughout this work, we mainly focus on the control of position and orientation of the arm's tip (d).

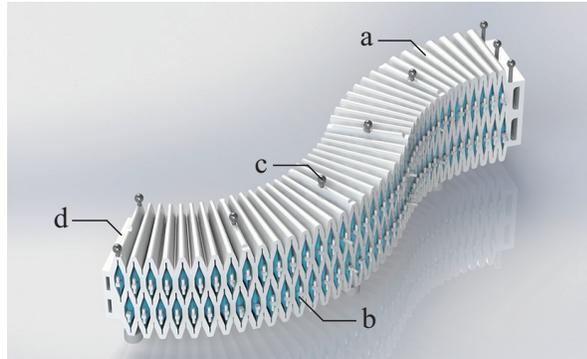


Fig. 2. The HPN arm composed of five segments, with a 3D-printed frame (a), airbags (b) and reflective markers (c), the arm's tip (d).

An individual segment is detailed in Fig. 3(B), containing eight connected chambers in one row, and there are two airbags arranged in one chamber side by side. So there are four groups of airbags in one segment. When each airbag group is pressurized separately, the whole segment bends towards the part with lower pressure. Specifically, when airbags on two levels are pressurized with different pressure (Fig. 3(C)), the segment bends in the y - z plane. Similarly, it also can bend in the x - y plane (Fig. 3(D)). Besides, it will elongate if airbags on both levels are pressurized together. In this work, we only focus on control for the HPN arm in x - y plane.

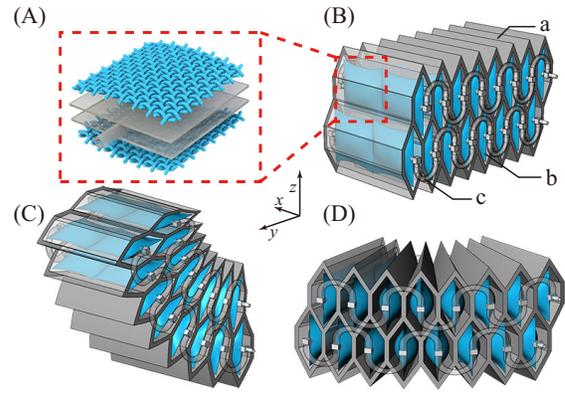


Fig. 3. The basic structure and working principle of a single segment of HPN arm. The segment (B) consists of a main frame (a), airbags (b) and S-shaped air tubes (c). An airbag is detailed in (A). The airbag consists of two layers of TPU film, and it is reinforced by processed nylon fabric. The airbag can be pressurized through one air tube with a maximum pressure of 0.5 Mpa. (C) and (D) show the bending deformations in two directions.

III. CONTROL

As mentioned before, the arm consists of several segments, each of which has identical chambers pressurized simultaneously. In addition, we assume that the deformations of different segments are independent, so it's proper to consider that the arm's backbone satisfies the piecewise constant curvature (PCC) assumption [17], as shown in Fig. 4.

The control problem can be described as the mapping from the task space to the actuation space of a robot. In our control algorithm, the problem is divided into two levels: from the desired arm tip's position and orientation in task space to the curvatures and arc lengths of all the arm segments in configuration space, and from the latter to the pressure of each airbag groups in actuation space. Pose optimization and neural-network method are used in these two levels respectively.

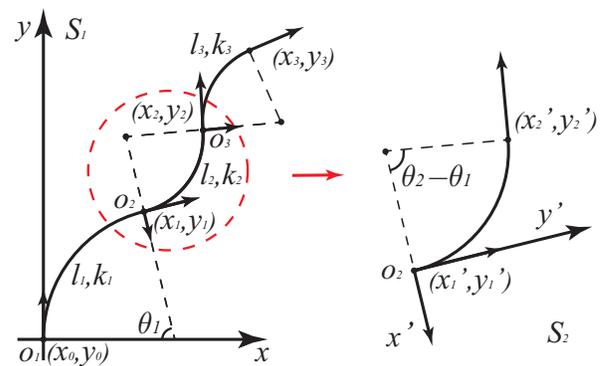


Fig. 4. The kinematics model of HPN arm is described in Cartesian coordinate system. (x_i, y_i) represents the coordinate in the global coordinate system while (x'_i, y'_i) represents the relative coordinate in the local coordinate system for segment i . k_i and l_i represent the curvature and arc length of segment i respectively, and θ_i represents the sum of first i segments' bending angle.

A. Parameter Estimation

In order to control the pose of the soft robot arm in task space, it is first necessary to estimate an arm segment's pose (curvature and arc length) in configuration space using available localization data. To formalize the problem, we build right-handed Cartesian coordinate systems S_1, S_2, \dots, S_n for each segment respectively, by fixing the positive direction of y-axis to the starting tangent vector of each segment's bending arc, as shown in Fig. 4. Specifically, S_i is the local coordinate system for segment i , while S_1 is also set as the global coordinate system.

As shown in Algorithm 1, for each segment i , we convert the global coordinate (x_i, y_i) in S_1 to local coordinate (x'_i, y'_i) in S_i , in order to figure out the curvature k_i and arc length l_i . And θ_i represents the angle between the tangent vector at the tip of segment i and y-axis in S_1 , which can be calculated by adding $k_i l_i$ to θ_{i-1} .

Algorithm 1 Parameter Estimation

```

1: function ESTIMATION( $\vec{X}, \vec{Y}, n$ ) ▷ n: segment number
2:    $k_1 \leftarrow \frac{2x_1}{x_1^2 + y_1^2}, \theta_1 \leftarrow 2 \arctan \frac{x_1}{y_1}, l_1 \leftarrow \frac{\theta_1}{k_1}$ 
3:   for  $i = 2, i \leq n, i++$  do
4:      $x'_i \leftarrow -\sin \theta_{i-1}(x_i - x_{i-1}) + \cos \theta_{i-1}(y_i - y_{i-1})$ 
5:      $y'_i \leftarrow \cos \theta_{i-1}(x_i - x_{i-1}) + \sin \theta_{i-1}(y_i - y_{i-1})$ 
6:      $k_i \leftarrow \frac{2x'_i}{x'^2_i + y'^2_i}, l_i \leftarrow \frac{2}{k_i} \arctan \frac{x'_i}{y'_i}$ 
7:      $\theta_i \leftarrow k_i l_i + \theta_{i-1}$ 
8:   end for
9: end function

```

B. Pose Optimization

Cause of redundant DoFs of soft arms, there are theoretically infinite poses in configuration space corresponding to a desired position of arm's tip in task space. Therefore, in order to select the optimal pose, a combined cost function containing three basic function is set for evaluating a certain pose (positions of all the segment tips). Each basic function is represented as a penalization cost term, which is proportional to the current pose's deviation from the preset requirements. Using gradient descent, an optimal pose under the measure can be figured out.

As shown in Algorithm 2, the tip position of all the segments, (\vec{X}, \vec{Y}) , are initialized and transformed to the curvatures and arc lengths (\vec{K}, \vec{L}) using Algorithm 1, where (x_t, y_t, θ_t) represents the position and orientation of target, and $k_{max}, k_{min}, k_{avg}, l_{max}, l_{min}, l_{avg}$ represent the maximum, minimum and average of curvature and arc length respectively, which can be measured in advance.

For the tip of each segment, the theoretical feasibility of reaching a certain position is considered in $Cost_1$, where f represents whether the combination of curvature k and arc length l of one segment are in the available range. If $f < 0$, the corresponding k, l are in the available range, and $Cost_1$ is close to 0. Otherwise, $Cost_1$ sharply grows to a great positive value, which implicates unavailable k, l . Besides, δ ($\delta \ll 1$) ensures smooth variance when f is close to 0.

The accuracy requirement of the arm tip's arrival orientation is considered in $Cost_2$, which grows to an relatively large value when there is a deviation, where θ_t, θ_r represent the orientation of the target and arm root respectively, and θ_n represents the total bending angle of first n segments.

$Cost_3$ ensures all segments in a certain pose with similar elongation rates and least sum of the absolute of bending angles because such poses are natural and adaptable. Specifically, the pose composed of evenly deformed segments provides space for further elongations and bends.

Finally, weighted with different empirical parameters according to the significance ($\alpha > \beta > \gamma$), three basic cost functions compose an integrated cost function $Cost$. Then, using *fminunc* in Matlab Toolbox, the optimal positions of each arm segment's tip (\vec{X}^*, \vec{Y}^*) can be figured out, and then transformed to (\vec{K}^*, \vec{L}^*) as the inputs of neural network detailed in next section.

Algorithm 2 Pose Optimization

```

1: function OPTIMIZATION( $x_t, y_t, \theta_t, n$ ) ▷ n: segment number
2:    $(\vec{X}, \vec{Y}) \leftarrow Initialization(x_t, y_t, \theta_t)$ 
3:    $(\vec{K}, \vec{L}) \leftarrow ESTIMATION(\vec{X}, \vec{Y}, n)$ 
4:    $f \leftarrow \left| \frac{2(l - l_{avg})}{l_{max} - l_{min}} \right| + \left| \frac{2(k - k_{avg})}{k_{max} - k_{min}} \right| - 1$ 
5:    $Cost_1 \leftarrow \sum_{i=1}^n \left( \sqrt{f_i^2 + \delta^2} + f_i \right)$ 
6:    $Cost_2 \leftarrow (\theta_t - \theta_r - \theta_n)^4$ 
7:    $Cost_3 \leftarrow \sum_{i=1}^n \left( \frac{l - l_{avg}}{l_{max} - l_{min}} \right)^2 + \left( \frac{k - k_{avg}}{k_{max} - k_{min}} \right)^2$ 
8:    $Cost \leftarrow \alpha Cost_1 + \beta Cost_2 + \gamma Cost_3$ 
9:    $(\vec{X}^*, \vec{Y}^*) \leftarrow fminunc(Cost, (\vec{X}, \vec{Y}))$  ▷ Matlab Toolbox
10:   $(\vec{K}^*, \vec{L}^*) \leftarrow ESTIMATION(\vec{X}^*, \vec{Y}^*, n)$ 
11: end function

```

C. Neural Network

Another critical component to the main control algorithm is the neural network, *Net*, that learns the relationship between the configuration space and the actuation space (pressure). Because different segments of HPN arm have different parameters, *Nets* are trained separately for each segment. For each *Net* corresponding to one segment, as shown is Fig. 5, we set the curvature k , arc length l as two inputs and the pressure p_l, p_r in airbag groups on the left and right side as two outputs.

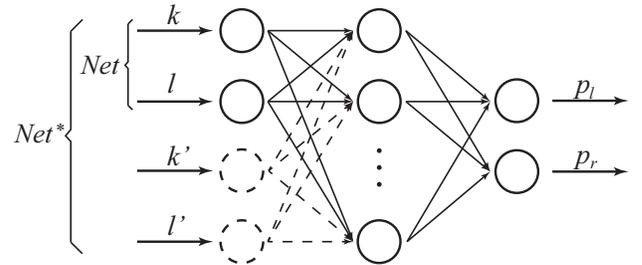


Fig. 5. Neural network *Net* is used to figure out the relationship between configuration space and actuation space with two inputs: curvature k and arc length l , and two outputs: pressures on each side, p_l, p_r . Advanced *Net** has four inputs, added the curvature k' and arc length l' of last pose, to compensate the effect of viscoelasticity.

In the training process of Net_i for the i th segment, we first generate enough training data by pressurizing the airbag groups using pressure samples (P_l, P_r) randomly selected in the available pressure range (P_{min}, P_{max}) and record the segment tip's positions (X, Y) and converted to (K, L) using Algorithm 1. Using the data pairs, $((K, L), (P_l, P_r))$, we can get Net_i using *train* function in Matlab Toolbox.

D. Main Control Algorithm

The main algorithm (Algorithm 3) is to find the relationship between the target's pose in task space and the pressure in actuation space. When a target's pose (x_t, y_t, θ_t) is given, we first calculate (\vec{K}, \vec{L}) for all segments using Algorithm 2, and then get each segment's pressure (\vec{P}_l, \vec{P}_r) from the corresponding trained Net . Finally, airbag groups are pressurized (\vec{P}_l, \vec{P}_r) to actuate the arm to reach the target.

Algorithm 3 Main Control Algorithm

```

1: function CONTROL( $x_t, y_t, \theta_t, n$ )           ▷ n: segment number
2:   Depressurization
3:    $(\vec{K}, \vec{L}) \leftarrow$  OPTIMIZATION( $x_t, y_t, \theta_t, n$ )
4:   for  $i = 1, i \leq n, i++$  do
5:      $(P_{li}, P_{ri}) \leftarrow Net_i(k_i, l_i)$ 
6:   end for
7:   Pressurization using  $\vec{P}_l, \vec{P}_r$ 
8: end function

```

E. Considering Viscoelastic Behavior

As we mentioned before, the viscoelasticity of soft material is a major factor affecting the open-loop control accuracy of soft robots. For our soft arm, same control parameters (pressure) can lead to different positions and orientations of the arm's tip affected by the arm's former poses. Accordingly, in Algorithm 3, all arm segments need to be depressurized in order to reduce the effect of viscoelasticity, which not only lengthens the execution time but also makes it impossible to implement consecutive movements using this control algorithm.

To reduce the effect of viscoelasticity, a new strategy considering the former poses of the arm is proposed. In pre-experiments, it's found that the last pose plays the main role in affecting the arm's current state among several former poses, and it's quite time-consuming to train a neural network with many former states as the input. Therefore, we only take the last pose into consideration. With input of the current and last pose, Net^* is designed to learn the relationship between configuration space and actuation space, as shown in Fig. 5 (with consideration of dotted part). During the training process of Net^* , the two poses are recorded and converted to k, l and k', l' as the input. Thus, Net^* can learn the relationship with consideration of viscoelastic behavior.

Since Net^* is well trained by a large amount of data, we use the estimated k', l' corresponding to the last target as its input instead of the real last pose performed by the arm. Which means the strategy can be used under open-loop environment, though it would be better to use real last pose as input. Thus, consecutive open-loop motion control with compensation for viscoelastic effect can be realized.

IV. EXPERIMENTS

A. Control System

The whole control system, as shown in Fig. 6, consists of computational device (A), analog voltage signal generator (WK-AO08)(B), air treatment device (SMC, AC30C-02DG-A)(C), proportional pressure regulator (SMC, ITV0030-2BL)(D) and a prototype of five-segment HPN arm (E) developed in [14].

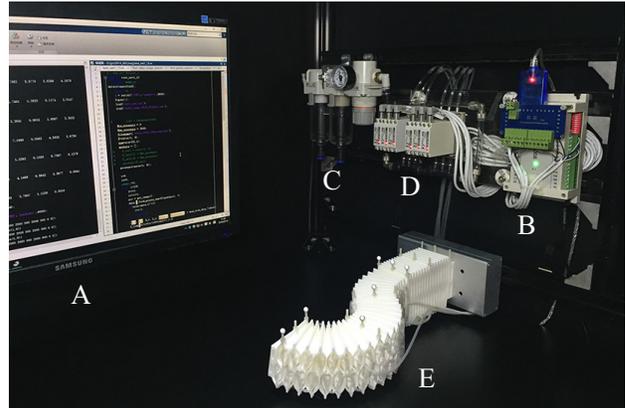


Fig. 6. Real control system including computational device (a), analog voltage signal generator (b), air treatment device (c), pressure proportional regulator (d) and an HPN arm (e).

The working process of the control system is detailed in Fig. 7. The air pump (a) generates an airflow with a pressure of about 0.7 Mpa. The airflow is preprocessed by the air treatment device (b) where the pressure is stabilized to 0.4 Mpa, and then directed to the proportional pressure regulators (c). Besides, the control signal from the computational device (d) is converted to analog voltage signal by the signal generator (e) and sent to pressure regulator to adjust the output pressure actuating the HPN arm (f). Besides, the positions of reflective markers on the HPN arm are recorded by a Kinect2 camera (not depicted) and sent to the computational device.

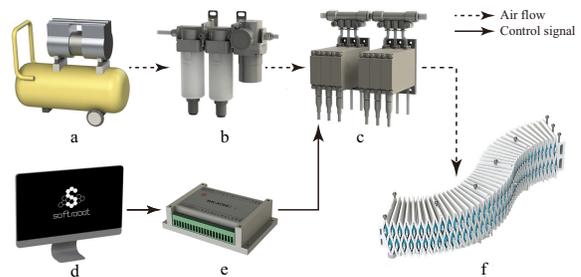


Fig. 7. Framework of control system: Control signal generated from the computational device (d) is converted to analog voltage signal by the signal generator (e) and passed to the proportional regulator (c), and it controls the output pressure; Air flow generated by the pump (a) is preprocessed by the air source treatment device (b) and input to the proportional regulator (c).

For a manipulation system, it's essential to do precise position and orientation control of the soft arm's tip. To achieve this goal, the single segment test is firstly implemented to

verify the neural network solution. Then, we implement the multi-segment test of the whole arm for the whole control algorithm. Moreover, feedback strategy is integrated to increase closed-loop control accuracy and a path tracking test is conducted to test the stability of the system.

B. Single Segment Test

A foundation for multi-segment control is that a single segment can be controlled to achieve the desired curvature and arc length. For single segment, according to the PCC model, its tip's orientation can be calculated by its tip's position, so only the position is concerned in this section. We select the second segment of the arm as an example, Fig. 8 details the positional errors.

We randomly generate pressure samples distributed in 0-0.3Mpa, and collect the positions of segment's tip. Using 2000 pairs of data (pressures and positions of segment's tip), we train the *Net* with 25 neurons in hidden layer. Its precision is tested using 100 random targets. Both in the training and testing process, the arm is initialized (fully deflated for 4 seconds) when an execution is finished. The result is shown in Fig. 8(a), with an average positional error of 2.38 mm.

Initialization process costs much time, so we then conduct a similar experiment to train another *Net* without initialization. The corresponding result is shown in Fig. 8(b), with an average error of 3.11 mm.

Then, we train *Net** using the same training samples and test set, whose process doesn't need initialization either. As shown in Fig. 8(c), it's obvious that *Net** provides a much better solution in precision, with an average error of 1.82mm. With consideration of viscoelasticity, the precision is improved by about 40%, compared with the counterpart using *Net* without initialization.

C. Multi-Segment Control

For multi-segment control, the first three segments of the HPN arm are used for precision test. Neural networks corresponding to the three segments are trained separately, similar to the process in the single segment test. 100 targets are randomly selected in the workspace, and used as input to the main control algorithm (Algorithm 3). The positional and orientational error distributions are detailed in Fig. 9. In Fig. 9(a), the arm is initialized (completely deflated) after reaching each target, where the mean positional and orientational error are 13.56mm and 4.39 degrees respectively. In Fig. 9(b), the test is without initialization process and the average errors grow to 17.54mm and 6.26 degrees. Then in Fig. 9(c), the test is still without initialization process but neural networks used in this test are supplanted by those with additional inputs of a former pose (*Net**). The corresponding results are 11.45mm and 4.04 degrees. It is obvious that the strategy handling viscoelasticity efficiently reduces the average positional and orientational errors. Moreover, as shown in the left one of Fig. 9(c), the decrease of maximum positional error implies the strategy also improves the control stability.

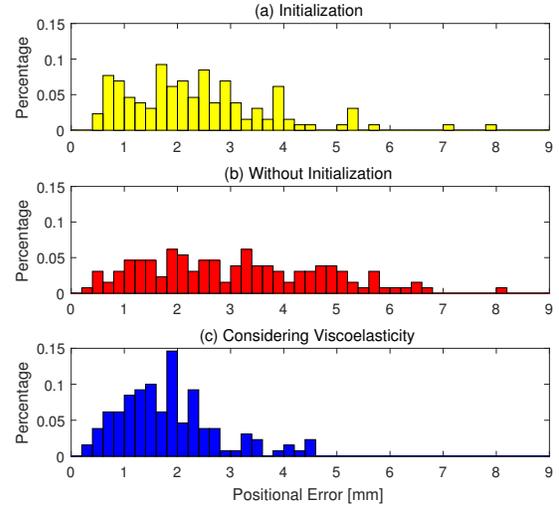


Fig. 8. Comparison of single segment's precision. (a) represents the error distribution when the arm is controlled to reach the given targets with initialization between two executions, where the mean error is 2.38 mm; the similar process is then conducted without initialization after each execution, whose error is shown in (b), with a mean of 3.11 mm; (c) shows the counterpart of the process using *Net**, whose mean error is 1.82 mm, much less than the prior two.

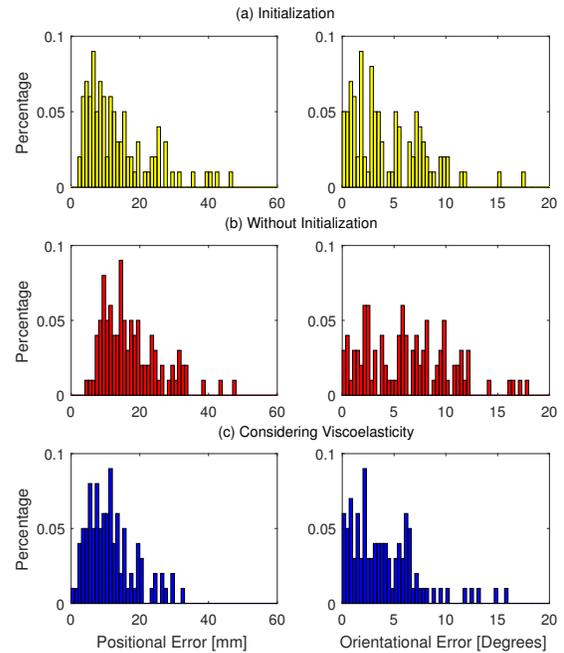


Fig. 9. Comparison of precision for the HPN arm with three segments. (a) represents the error distributions when the arm is consecutively controlled to the given poses with initialization, and its mean positional error is 13.56 mm, and orientational error is 4.39 degrees; the similar process is then conducted without initialization after each execution, whose errors are shown in (b), with a mean positional error of 17.54 mm, and orientational error of 6.26 degrees; (c) shows the counterpart of the process using *Net**, whose mean errors are 11.45 mm and 4.04 degrees, less than the prior two.

D. Combination with Feedback Strategy

When the actuator is used in an application scene with no restrictions of sensors or response time, the feedback strategy can further improve the precision. We use a scheme that integrates the main control algorithm with additional feedback controller. Specifically, we use the error between the desired pose $q = (x, y, \theta)$ and real pose $q' = (x', y', \theta')$ as a modification term of the input $q^* = (x^*, y^*, \theta^*)$ to the control system in the iteration process:

$$q^* \leftarrow q^* + \alpha(q - q')$$

Where α represents the modification rate. The iteration process ends when the error falls into an acceptable range, or the number of iterations reach a limit (set as 10 in practice). The process contains 5 iterations on average, each taking 1s for data gathering, calculation and actuation.

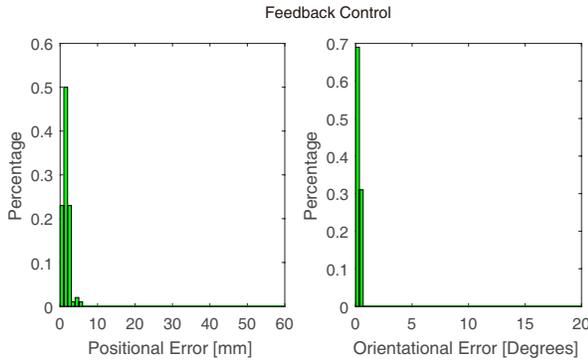


Fig. 10. Errors of feedback control. The mean positional error is 1.61 mm while the mean orientational error is 0.26° .

As shown in Fig. 10, with feedback strategy, the mean positional and orientational errors of 100 trials are 1.61 mm and 0.26° respectively, much less than those in former open-loop control.

E. Path Tracking

The path tracking experiment is conducted to test the precision and stability of the control system. This problem is decomposed to numbers of subtasks: reach numerical targets distributed on the path. As the targets are distributed closely, instead of iteration over each point, we propose a much more efficient strategy dynamically compensating for the previous errors: the error in the i th subtask is inherited to the determination of the $(i+1)$ th target, which can be represented as:

$$q_{i+1}^* \leftarrow q_i^* + (q_{i+1} - q_i) + \beta(q_i - q_i'), \quad i \geq 1$$

Where $q_i^* = (x_i, y_i, \theta_i)$ represents the i th input to the control system, and q_i, q_i' represent the i th desired pose and real pose respectively, while β represents the modification rate.

In real experiment, we set β as 1, so we get:

$$q_{i+1}^* \leftarrow q_i^* + q_{i+1}' - q_i, \quad i \geq 1$$

Fig. 11 shows several arm's measured pose at the experiment's start (O), path's start (A), path's turning point (B)

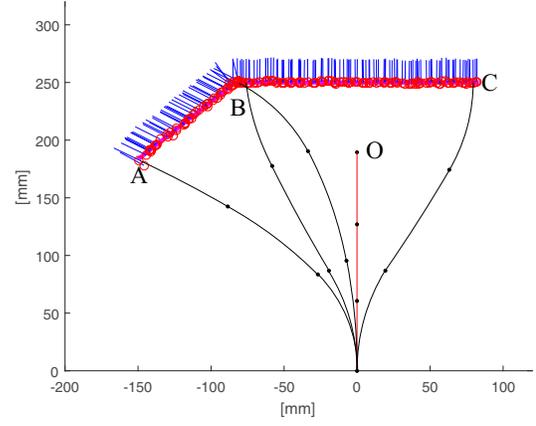


Fig. 11. The experimental trial of path tracking program. The target path is shown in magenta. The inclined line, with an angle of $\frac{\pi}{4}$, has a target orientation of $\frac{\pi}{3}$ relative to y-axis, and the horizontal line has a target orientation parallel to y-axis. The arm has an original pose O, and the path's start A, end C and turning point B. The red circle represents each step's position, while the blue line represents corresponding measured orientation.

and path's end (C). The measured arm tip's pose is shown at each step along the path, where the position and orientation are represented as red circles and blue lines respectively. The error result is shown in Fig. 12 with a mean positional and orientational error of 1.52 mm and 0.43° . The orientational error sharply grows at the turning point (B) because the arm has a huge state change.

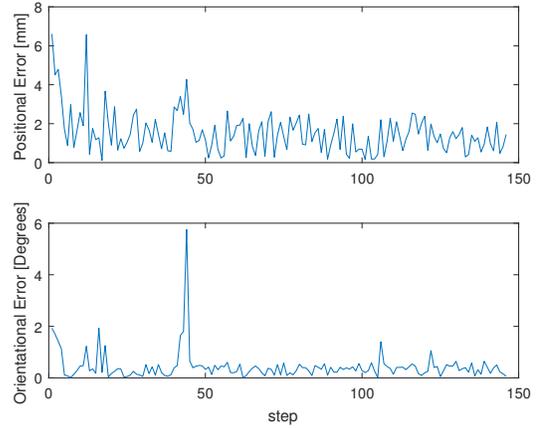


Fig. 12. Error records in path tracking. The positional error is relatively high at the beginning and the turning point. The mean positional error is 1.52 mm. The orientational error sharply grows at the turning point. The mean orientational error is 0.43° .

From the track recorded in the experiment, we can see that an arm with a large extension ratio is able to finish a task in a natural way, which means it can move in combination of elongation and smooth bending when reaching a target. In contrast, an inextensible one may have a large reachable space, but it needs to bend large angles to reach a close target on its side.

V. DISCUSSION

Our algorithm successfully settles the problem of positional and orientational control for the HPN multi-segment soft extensible arm. However, there are still several limitations:

- Only the last pose of the segment is taken as an additional input of the neural-network (Net^*). However, it would be better to take more former poses into consideration if the training time is acceptable.
- This algorithm's application range is now restricted in 2D plane. In future work, enhanced neural networks considering the gravity effect of external load and arm itself can be designed in order to extend the algorithm to 3D space.
- Differing from those soft arms with rigid plates between segments [18], the HPN arm is an uniform one without obvious boundary between two segments. Therefore, it is not reasonable to assume the deformation of each segment is independent as mentioned in Section III. This may be the reason why the precision of multi-segment control is not as satisfactory as that in single segment test.

VI. CONCLUSION

This paper outlines a control algorithm for a multi-segment extensible HPN arm. The inverse kinematics is resolved into two levels: from task space to configuration space, and from configuration space to actuation space. In the first level, we use gradient descent to figure out optimized positions of all segment tips according to preset measures. Neural-network considering the viscoelasticity effect is used in the second level, which gives accurate pressure outputs from the arc length and curvature of a certain segment. The highlight of this control method also lies on hierarchical strategy, which greatly simplifies the problem for each level. Specifically, the neural networks employed in this work only handle the situation of single segment, which greatly reduce the amount of training data. Besides, the feedback control implemented in our control system achieves a great precision in prototype experiments. In the future, control in 3D space and path planning will be considered. The accompanying video provides a demonstration of the proposed control approach.

REFERENCES

- [1] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.
- [2] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: a bioinspired evolution in robotics," *Trends in biotechnology*, vol. 31, no. 5, pp. 287–294, 2013.
- [3] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Applied Bionics and Biomechanics*, vol. 5, no. 3, pp. 99–117, 2008.
- [4] R. F. Shepherd, F. Ilievski, W. Choi, S. A. Morin, A. A. Stokes, A. D. Mazzeo, X. Chen, M. Wang, and G. M. Whitesides, "Multigait soft robot," *Proceedings of the National Academy of Sciences*, vol. 108, no. 51, pp. 20400–20403, 2011.
- [5] T. Mahl, A. Hildebrandt, and O. Sawodny, "A variable curvature continuum kinematics for kinematic control of the bionic handling assistant," *IEEE transactions on robotics*, vol. 30, no. 4, pp. 935–949, 2014.

- [6] F. Chen, M. Y. Wang, J. Zhu, and Y. Zhang, "Interactions between dielectric elastomer actuators and soft bodies," *Soft Robotics*, vol. 3, no. 4, pp. 161–169, 2016.
- [7] I. S. Godage, E. Guglielmino, D. T. Branson, G. A. Medrano-Cerda, and D. G. Caldwell, "Novel modal approach for kinematics of multisection continuum arms," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 1093–1098.
- [8] I. S. Godage, D. T. Branson, E. Guglielmino, G. A. Medrano-Cerda, and D. G. Caldwell, "Shape function-based kinematics and dynamics for variable length continuum robotic arms," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 452–457.
- [9] C. Escande, R. Merzouki, P. M. Pathak, and V. Coelen, "Geometric modelling of multisection bionic manipulator: Experimental validation on robotinoxt," in *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2006–2011.
- [10] M. Giorelli, F. Renda, G. Ferri, and C. Laschi, "A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three-dimensional space," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 5033–5039.
- [11] 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 Transactions on Robotics*, vol. 31, no. 4, pp. 823–834, 2015.
- [12] D. Braganza, D. M. Dawson, I. D. Walker, and N. Nath, "A neural network controller for continuum robots," *IEEE transactions on robotics*, vol. 23, no. 6, pp. 1270–1277, 2007.
- [13] M. Rolf and J. J. Steil, "Efficient exploratory learning of inverse kinematics on a bionic elephant trunk," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 6, pp. 1147–1160, 2014.
- [14] H. Jiang, X. Liu, X. Chen, Z. Wang, Y. Jin, and X. Chen, "Design and simulation analysis of a soft manipulator based on honeycomb pneumatic networks," in *Robotics and Biomimetics (ROBIO), 2016 IEEE International Conference on*, 2016. Accepted.
- [15] H. Sun and X.-P. Chen, "Towards honeycomb pneunets robots," in *Robot Intelligence Technology and Applications 2*. Springer, 2014, pp. 331–340.
- [16] H. Sun, N.-Y. Wang, H. Jiang, and X.-P. Chen, "Flexible honeycomb pneunets robot," *International Journal of Robotics and Automation*, vol. 31, no. 6, 2016.
- [17] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, 2010.
- [18] M. D. Grissom, V. Chitrakaran, D. Dienno, M. Csencits, M. Pritts, B. Jones, W. McMahan, D. Dawson, C. Rahn, and I. Walker, "Design and experimental testing of the octarm soft robot manipulator," in *Defense and Security Symposium*. International Society for Optics and Photonics, 2006, pp. 62 301F–62 301F.