Model-Less Feedback Control for Soft Manipulators

Yusong Jin[1], Yufei Wang[2], Xiaotong Chen[2], Zhanchi Wang[3], Xinghua Liu[3], Hao Jiang[2], and Xiaoping Chen[2]

***Abstract*— Soft manipulators have been a rising focus of soft robotics research. Taking advantage of soft materials and flexible, continuous movements, they have promising applicable prospect. However, their highly internal nonlinearity and unpredictable deformation caused by environmental effects make it difficult to build an exact model for control. In this work, we propose a generalized controller for soft manipulators using an estimated Jacobian-based model derived from structural analysis. The model can be simplified from reasonable assumptions of manipulator structure, and updated to balance conformity to reality and stability. In prototype experiments on an 3D multi-segment soft manipulator, the control method exhibits accuracy as well as adaptability to self gravity and external loads.**

I. INTRODUCTION

Compared with traditional rigid robots, soft robots can serve as better solutions in human-centric environments where safety and adaptability of uncertainty are fundamental requirements, as reviewed by Rus and Tolley [1]. Utilizing their continuous deformation and theoretically infinite degrees of freedom (DoFs) [2], they can adaptively grasp and manipulate unknown objects varying in size and shape [3], or squeeze through confined space [4]. Nevertheless, because soft robots are highly nonlinear and susceptible to environmental effects, it is a challenge to implement accurate control.

Control methods of soft manipulators can be divided into two categories, open-loop methods and closed-loop methods. Regarding open-loop methods, a relatively accurate model needs to be established. Zheng et al. [5] propose a 3D dynamic model for an octopus-arm like continuum robot with a multi-segment structure. Giorelli et al. [6] implement non-constant curvature control based on kinematics model. In fact, it is difficult to develop mathematical modeling for soft manipulators, due to their internal unidealities (hysteresis, backlash, etc.). Learning based methods can provide a more accurate model after effective training. Giorelli et al. [6] also develop control method using neural networks. Rolf et al. [7] implement 3D space control of the bionic handling assistant (BHA) using a novel learning based method called goal babbling. Jiang et al. [8] combine modeling and neural network for the control of a multi-segment extensible manipulator. Braganza et al. [9] present a neural network based method for dynamic modeling of OctArm with feed-forward components to compensate for model uncertainty. Generally, open-loop strategy can to some extent solve the problem of control for soft robots. However, these methods cannot reach high accuracy due to soft manipulators' unpredictability. Specifically, the compliance of the robot body creates infinite DoFs, and unsensed environmental effects (gravity, obstacles, external forces, etc.) make it impossible to implement accurate modeling.

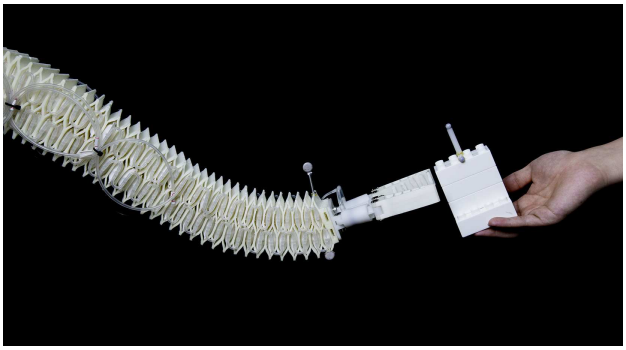

Fig. 1. The soft manipulator with a gripper is trying to catch an object.

Consequently, some contributions attempt to employ closed-loop strategy to improve accuracy. In fact, aforementioned works [7][8] integrate a simple feedback strategy onto the open-loop control system and achieve much higher accuracy. Wang et al. [10] develop adaptive controller for end effector positioning based on visual servo. Besides, some contributions implement feedback strategy in part of their control systems. Rolf et al. [7] use length sensors instead of cameras to directly adjust robot configurations in closed-loop control. Marchese et al. [11] implement closed-loop control of curvature for a 2D soft manipulator under the piecewise constant curvature assumption. These methods rely on model accuracy in feedback control, which would lead to failure when the real manipulator is not conformed to models due to unidealities as well as external environmental effects, which cannot be wholly sensed on soft continuum manipulators.

Moreover, several model-free feedback control methods are introduced in previous researches. Kormushev et al. [12] develop kinematic-free control of a rigid arm by local model from generating actuation primitives and perceiving their effect on end effector. Yip et al. [13] implement model-less position control of a 2D cable-actuated manipulator by online updating of an estimated model during feedback loop, and extend it to hybrid position-force control in [14]. These works show that feedback control is feasible without

*This research is supported by the National Natural Science Foundation of China under grant 61573333.

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

[2]Hao Jiang, Xiaotong Chen, Yufei Wang 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

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

explicit model, nevertheless, these methods may fail when the manipulator configuration goes more complicated that cannot be reasonably estimated by measurement and online updating. In other words, without substantial model analysis, the feedback loop cannot ensure global convergence.

In this work, we propose a control method based on the model-less concept: control based on an estimated model built offline and consecutively updated during feedback process. Our estimated model is built from approximate kinematics analysis rather than totally measurement, which ensures stability in workspace and effective feedback, and updating strategy conforms the estimated model to reality as well as stability and convergence requirement. The controller is implemented on a novel soft manipulator developed by authors in [15] (shown in Fig. 1). Specifically, feedback parameters are positional, directional descriptors of end effector and air pressure for actuation, and parameters representing their changing ratio in Jacobian matrix are simplified according to structural analysis. The method shows adaptability to self gravity as well as external loads with high accuracy in positioning and path tracking experiments. To the best of our knowledge, this work is the first implementation of accurate position and direction control in 3D space of soft manipulators without explicit model, and it's a general method for other soft manipulators.

II. METHOD

In this section, we introduce the feedback control method based on an estimated Jacobian model, which is implemented on a novel manipulator called honeycomb pneumatic networks manipulator (HPN) [15]. With the target as reaching desired end effector's position and direction, the manipulator's structure and its deformation mode are analyzed for selecting appropriate variables as representatives in actuation space and task space. Based on the analysis, the Jacobian matrix is built and simplified, and its updating strategy ensures conformity to reality as well as convergence and stability. The controller framework is detailed in Fig. 2.

A. HPN Manipulator Structure

The HPN manipulator's structure is shown in Fig. 3, composed of five segments. Each segment has a soft frame and four groups of inflatable airbags inside for actuation. With assumption of segment independence, the manipulator's movement can be regarded as simple combination of each segment's movement. With assumption of no torsion around z-axis, a segment's movement can be decomposed into three basic motions: bending in x-z plane, in y-z plane, and elongation-retraction. Specifically, when airbags in group 1, 2 are inflated with the same pressure, and group 3, 4 are inflated with another same pressure, the segment will bend in y-z plane. Similarly, it bends in x-z plane when airbag groups 1, 3 and 2, 4 are inflated with the same pressures, respectively. And it elongates or retracts when all the airbags are inflated or deflated together with the same pressure. For short, later we will only mention elongation on behalf of elongation-retraction.

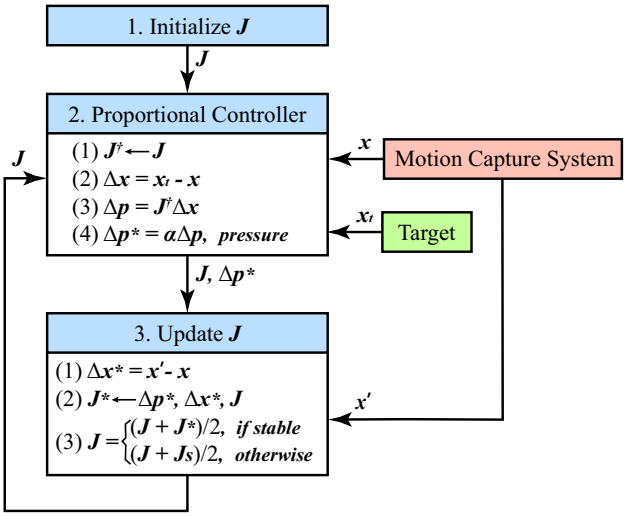

Fig. 2. The figure shows the framework of a proportional feedback controller based on estimated Jacobian matrix $\boldsymbol{J}$. During the feedback loop, the actuation change $\Delta\boldsymbol{p}$ is calculated from task space distance $\Delta\boldsymbol{x}$ and pseudo-inverse of Jacobian $\boldsymbol{J}^{\dagger}$, and scaled by damping ratio α to get real change $\Delta\boldsymbol{p}^{*}$. For update of $\boldsymbol{J}$, matrix $\boldsymbol{J}^{*}$ is derived from real task space change $\Delta\boldsymbol{x}^{*}$, and $\boldsymbol{J}$ will be replaced according to the stable condition, which makes $\boldsymbol{J}$ conform to $\boldsymbol{J}^{*}$ when manipulator movement is considered stable, or a safe value $\boldsymbol{J}_{s}$ ensuring stability and convergence.

B. Jacobian Matrix

1) Actuation Space Variables: As analyzed above, the manipulator's movement can be regarded as combination of three modes of motion, so we need to choose appropriate pressure combinations corresponding to the motions. Thus, for segment i, we define three pressure variables as:

$$\begin{cases} p_{ix} = -p_{i1} + p_{i2} - p_{i3} + p_{i4} \\ p_{iy} = \ \ p_{i1} + p_{i2} - p_{i3} - p_{i4} \\ p_{iz} = \ \ p_{i1} + p_{i2} + p_{i3} + p_{i4} \end{cases} \tag{1}$$

where p_{ix}, p_{iy} and p_{iz} correspond to bending in x-z plane, y-z plane and elongation, respectively, and $p_{i1}, \ldots, p_{i4}$ are pressures of airbag groups for segment i, as shown in Fig. 3. Notice that there exist infinite solutions for (1), so we simply

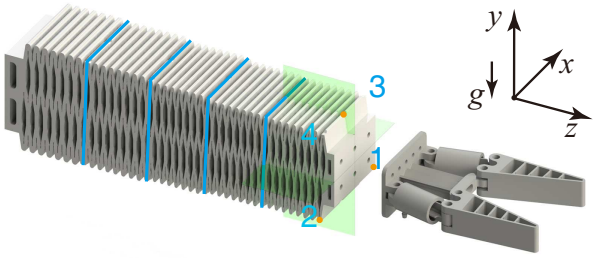

Fig. 3. The figure shows the structure of the HPN manipulator. The layout of airbags in the first segment is marked. Each segment has four groups of airbags. The base coordinate system is defined with y-axis along gravity acceleration, and z-axis along the initial shape of the manipulator.

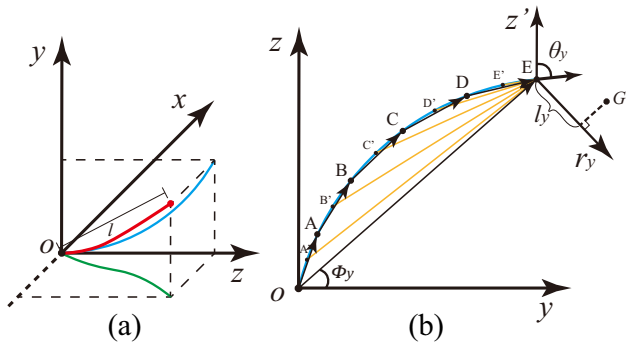

Fig. 4. (a) illustrates the manipulator's backbone curve in 3D space, and (b) illustrates its projection on y-z plane. In (a), red curve is the 3D backbone curve, blue and green curves are the projections; l is the length of cuboid diagonal, linking the base and end effector of manipulator. In (b), axis r_y is orthogonal to the vector from base to end effector, $\vec{OE}$. z' is parallel to z. A-E are segments' tips and A'-E' are segments' middle points, from which virtual rotation rods (yellow lines) are connected to the end effector E. G is the target's projection. l_y is displacement of G to E along r_y, and θ_y is the angle between z' and end effector tangent.

select $p_{i1},...,p_{i4}$ as:

$$\begin{cases} p_{i1} = (p_{iz} - p_{ix} + p_{iy})/4 \\ p_{i2} = (p_{iz} + p_{ix} + p_{iy})/4 \\ p_{i3} = (p_{iz} - p_{ix} - p_{iy})/4 \\ p_{i4} = (p_{iz} + p_{ix} - p_{iy})/4 \end{cases} \tag{2}$$

2) Task Space Variables: For task space definition, due to similarity of motions in x-z plane and y-z plane, we just consider the projection of manipulator's backbone in y-z plane, as shown in Fig. 4. We construct another axis r_y at end effector E, orthogonal to $\vec{OE}$, the vector from base to end effector.

As for bending motion of each segment, we assume its result for the front part from this segment to end effector can be approximated as the rotation of a virtual rigid rod linking the segment's middle point and end effector (in Fig. 4, when $\vec{OA}$ bends, part A to E will deform as rotation of rod $\vec{A'E}$). The positional change caused by rotation is approximated to l_y, the projection on axis r_y, and we define it as positional variable. Besides, we define θ_y, the angle between tangent at end effector and z axis, as directional variable. Similarly, we define l_x and θ_x as variables on x-z plane.

As for elongations of each segment, their directions are approximated to the vectors from segment base to tip ($\vec{OA}$, $\vec{AB}$, etc.). With respect to the whole manipulator's elongation in y-z plane, $\vec{OE}$ can roughly be the average direction. Due to similarity on two planes, instead of two projected lengths, we simply define the cuboid diagonal length l as variable for elongation.

As we get $[x,y,z,\theta_x,\theta_y]$, the original position and direction information from motion capture system during the feedback control, we need to convert $[x,y,z]$ to $[l_x,l_y,l]$, which can be easily derived as:

$$\begin{cases} l = \sqrt{x^2+y^2+z^2} \\ l_x = (x^*-x)\sin\phi_x - (z^*-z)\cos\phi_x \\ l_y = (y^*-y)\sin\phi_y - (z^*-z)\cos\phi_y \end{cases} \tag{3}$$

where x,y,z and x^*,y^*,z^* are end effector E and target G's coordinates in base coordinate system, and ϕ_y is shown in Fig. 3(b) as angle between $\vec{OE}$ and y-axis, same as ϕ_x. θ_x and θ_y can be directly recorded and do not need conversion.

3) Jacobian: Thus, we have defined variable representatives, $\boldsymbol{p} = [p_{1x},p_{1y},p_{1z},...,p_{5z}]^T$, $\boldsymbol{x} = [l_x,\theta_x,l_y,\theta_y,l]^T$ in actuation space and task space, so the Jacobian matrix $\boldsymbol{J} \in \mathbb{R}^{5\times 15}$ can be expressed as:

$$\boldsymbol{J} = \left(\frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{p}_j}\right)_{i,j} = \begin{bmatrix} \frac{\partial\theta_x}{\partial p_{1x}} & \frac{\partial\theta_x}{\partial p_{1y}} & \frac{\partial\theta_x}{\partial p_{1z}} & \cdots & \frac{\partial\theta_x}{\partial p_{5z}} \\ \frac{\partial l_x}{\partial p_{1x}} & \frac{\partial l_x}{\partial p_{1y}} & \frac{\partial l_x}{\partial p_{1z}} & \cdots & \frac{\partial l_x}{\partial p_{5z}} \\ \frac{\partial\theta_y}{\partial p_{1x}} & \frac{\partial\theta_y}{\partial p_{1y}} & \frac{\partial\theta_y}{\partial p_{1z}} & \cdots & \frac{\partial\theta_y}{\partial p_{5z}} \\ \frac{\partial l_y}{\partial p_{1x}} & \frac{\partial l_y}{\partial p_{1y}} & \frac{\partial l_y}{\partial p_{1z}} & \cdots & \frac{\partial l_y}{\partial p_{5z}} \\ \frac{\partial l}{\partial p_{1x}} & \frac{\partial l}{\partial p_{1y}} & \frac{\partial l}{\partial p_{1z}} & \cdots & \frac{\partial l}{\partial p_{5z}} \end{bmatrix}$$

4) Simplification: In this part, we figure out terms in $\boldsymbol{J}$ from several assumptions.

We assume p_{ix},p_{iy} and p_{iz} for segment i, will only cause corresponding variables to change. Specifically, for p_{ix}, we assume that it will only cause bending motion in x-z plane, so $\frac{\partial\theta_y}{\partial p_{ix}}, \frac{\partial l_y}{\partial p_{ix}}$ and $\frac{\partial l}{\partial p_{ix}}$ will be zero, similarly, $\frac{\partial\theta_x}{\partial p_{iy}}, \frac{\partial l_x}{\partial p_{iy}}$ and $\frac{\partial l}{\partial p_{iy}}$ will also be zero. Similarly, we assume p_{iz} will not cause change except for l, so $\frac{\partial\theta_x}{\partial p_{iz}}, \frac{\partial l_x}{\partial p_{iz}}, \frac{\partial\theta_y}{\partial p_{iz}}$ and $\frac{\partial l_y}{\partial p_{iz}}$ will be zero.

As for $\frac{\partial\theta_x}{\partial p_{ix}}$ and $\frac{\partial\theta_y}{\partial p_{iy}}$, each segment's directional change contributes evenly to the end effector's, so we use the same parameters for all segments, further, we use constants $k_{\theta x}, k_{\theta y}$ to represent average effect of the ratio of changing velocity of θ_x to p_{ix}. As for $\frac{\partial l_x}{\partial p_{ix}}$ and $\frac{\partial l_y}{\partial p_{iy}}$, we assume that the tangential displacements, l_{ix}, l_{iy}, are proportional to the length of virtual rotation rods in Fig. 4, and we approximate their lengths to those at initial state (with all segments straight, along z-axis), so their ratio will be roughly $1:3:5:7:9$. Thus, $\frac{\partial l_x}{\partial p_{ix}}$ and $\frac{\partial l_y}{\partial p_{iy}}$ will be $(2i-1)k_{lx}$ and $(2i-1)k_{ly}$. As for $\frac{\partial l}{\partial p_{iz}}$, we also assume that each segment will contribute evenly to the length of whole manipulator, so we use k_l for all segments' estimation. Notice that k_{lx}, k_{ly} and k_l also constants representing the average ratio of changing velocity of l_x, l_y, l to p_{ix}, p_{iy}, p_{iz}, respectively.

Thus, the simplified estimated Jacobian will be:

$$\boldsymbol{J}=\begin{bmatrix} k_{\theta x} & 0 & 0 & k_{\theta x} & 0 & 0 & \cdots & k_{\theta x} & 0 & 0 \\ k_{lx} & 0 & 0 & 3k_{lx} & 0 & 0 & \cdots & 9k_{lx} & 0 & 0 \\ 0 & k_{\theta y} & 0 & 0 & k_{\theta y} & 0 & \cdots & 0 & k_{\theta y} & 0 \\ 0 & k_{ly} & 0 & 0 & 3k_{ly} & 0 & \cdots & 0 & 9k_{ly} & 0 \\ 0 & 0 & k_l & 0 & 0 & k_l & \cdots & 0 & 0 & k_l \end{bmatrix}$$

5) Initialization and Update: We initial the parameters in $\boldsymbol{J}$ as the ratio of available range of variables in task space and actuation space, specifically, available airbag pressure, position and direction ranges are pre-measured and convert to those of $\boldsymbol{p}$ and $\boldsymbol{x}$ according to (1), (3). During the feedback process, actual change $\Delta\boldsymbol{x}^* = [\Delta l_x^*, \Delta\theta_x^*, \Delta l_y^*, \Delta\theta_y^*, \Delta l^*]^T$, $\Delta\boldsymbol{p}^* = [\Delta p_{1x}^*, \Delta p_{1y}^*, \Delta p_{1z}^*, ..., \Delta p_{5z}^*]^T$ are recorded and calculated in each step, from which we can get real Jacobian $\boldsymbol{J}^*$ as follows:

$$\begin{cases} \boldsymbol{K} = [k_{\theta x}, k_{lx}, k_{\theta y}, k_{ly}, k_l]^T \\ \boldsymbol{K}^* = \dfrac{\Delta\boldsymbol{x}^*}{\boldsymbol{J}\Delta\boldsymbol{p}^*} \cdot \boldsymbol{K} \\ \boldsymbol{J}^* \leftarrow \boldsymbol{K}^* \end{cases} \tag{4}$$

where $\boldsymbol{K}$, $\boldsymbol{K}^*$ represent estimated and real parameter vectors. Notice that operations are conducted element-wise in the second equation.

On the other hand, we determine a safe Jacobian $\boldsymbol{J}_s$ that ensures stability and convergence. To get $\boldsymbol{J}_s$, parameters of Jacobian matrix are enlarged to preset values in order to reduce the pressure change in steps, which may cause errors to increase or oscillate during iteration. We develop the following strategy for iteration: when $\boldsymbol{J}^*$ derived in two consecutive steps are almost the same, we assume that manipulator's movement is stable and $\boldsymbol{J}^*$ is reliable, and update $\boldsymbol{J}$ to $(\boldsymbol{J}+\boldsymbol{J}^*)/2$, or it will be updated to $(\boldsymbol{J}+\boldsymbol{J}_s)/2$.

III. EXPERIMENT

A. Control System

The whole control system is illustrated in Fig. 5. The airflow, about 0.7 Mpa, is generated from the pump (a) and pre-processed by the air treatment device (e) and stabilized to 0.3 Mpa, and then directed to the proportional pressure regulators (f). The control signal from the computational device (c) is converted to analog voltage signal by the programmable logic controller (b) and sent to pressure regulator to adjust the output pressure actuating the HPN manipulator (g). Motion capture system (d) records the real-time position and direction information of reflective optical markers on the HPN manipulator (g) and target, and send that to the computational device (c).

B. Tests on the Property of HPN Arm

For testing the property of HPN arm, firstly we inflate all groups of airbags respectively by constant pressure. In this test, to make the arm horizontal, the pressures of airbags below are less than those of upper ones, and the groups of airbags symmetric to each other by y-z plane have exact the same pressure. of coordinate variations are illustrated in Fig. 6.

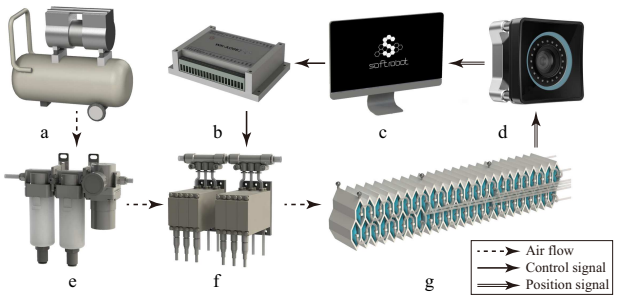

Fig. 5. The control system consists of air pump (a), analog voltage signal generator (WK-AO08) (b), computational device (c), motion capture system (OptiTrack Prime 17W×12) (d), airflow equipment (SMC, AC30C-02DG-A) (e), proportional pressure regulator (SMC, ITV0030-2BL) (f) and the 3D-printed HPN arm (g) using a kind of elastomer material (Polyflex). The transmission of air flow, control and position signals are illustrated by different arrows.

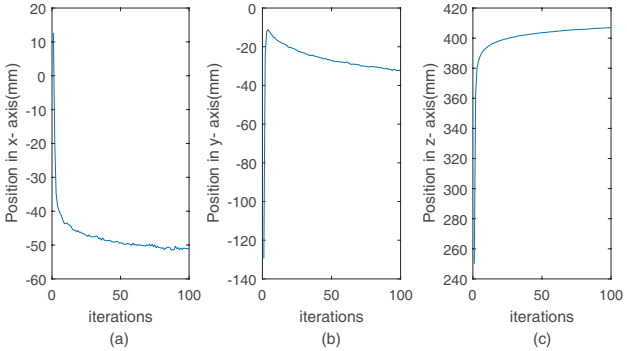

Fig. 6. The figure shows the variation of end effector position after inflation. x, y, z coordinate variations are shown in (a), (b), (c).

From Fig. 6, it can be found that the position of end effector changes violently in the first few seconds to reach a certain target, then the coordinate varies gradually. Besides, though the pressure of airbags which are symmetric to each other by y-z plane is the same, actually the end effector shifts to the right side while it shifts to the left at the beginning, according to Fig. 6(a). The reason is that the wall of air chambers in right side of the arm is a little thicker than the left due to manufacturing error. It's a common problem in the manufacturing process of soft manipulator. From Fig. 6(b) and Fig. 6(c), we can infer that the end effector rises up in 3 seconds and then declines while the arm elongates due to hysteresis effect of polymer. All of these phenomenons raise requirement of closed-loop control for soft manipulators.

Then we inflate the groups of airbags below and load the end effector with a 120g object. The end effector position variation in y-axis is illustrated in Fig. 7. It can be figured out that the end effector declines violently about 100mm due to external load, which means the material is extremely soft, which aggravates the difficulty of achieving a feasible control algorithm on our manipulator.

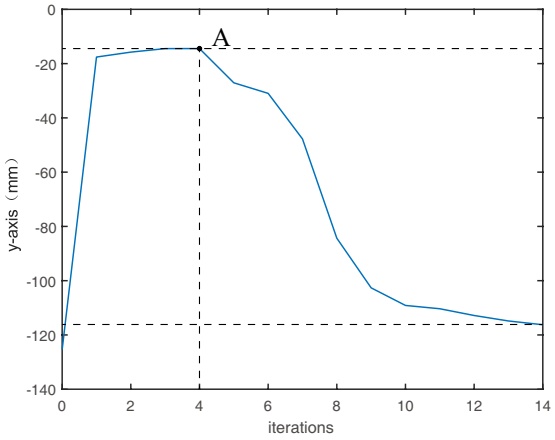

Fig. 7. The figure illustrates the position variation of end effector in y-axis in the process of inflation. At point A, a 120g object is loaded on the end effector,

C. Point-to-point Under Different Loads

The reaching performance in point-to-point test is shown in Fig. 8, where we can find that the three targets are gradually reached during the feedback control process. In this test, the end effector is only set to move in the x-y plane while keeping the distance in z-axis stable. From the trajectory records, we can see that the deviations are successfully corrected in the feedback process and do not influence accuracy. Besides, the end effector behaves to exceed the targets and then returns, signing the correction of hysteresis effect of material.

Besides, 30 tasks are set by randomly selecting targets in the manipulator workspace, and each is executed in 40 feedback iterations. Average error convergence is shown in Fig. 9. In Fig. 9(a) it can be recognized that the convergence rate is fast and stable, where the error converges to about 5mm in 15 iterations. Error bounce occurs at the 25th iteration due to the hysteresis effect. When the manipulator is under 100g load, the convergence rate is slightly slower yet negligible. Thus, it can be concluded that our algorithm adapts well to the load variation in the physical test. In addition, the algorithm is efficient when the error converges to about 5mm, yet inefficient to reduce the error.

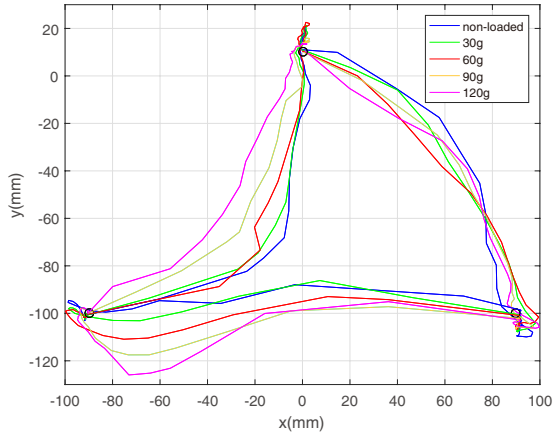

Fig. 8. The reaching performance in x-y plane of the manipulator in point-to-point reaching test to three targets under a load of 0, 30g, 60g, 90g and 120g respectively.

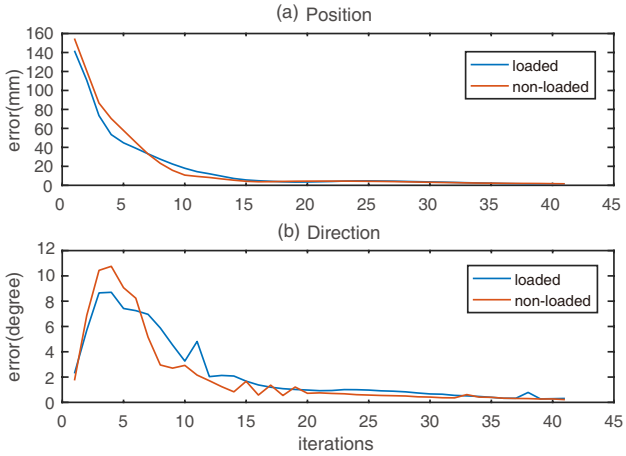

Fig. 9. The errors in position and direction during the point-to-point experiment are respectively illustrated in (a) and (b), with no load and 100g load. It is obvious that the errors converge quickly to less than 10mm and 2°.

In other words, reaching directions of selected targets are fixed at horizontal forth. It's shown in the Fig. 9(b) that the maximum average error is less than 11°, which converges to less than 4° in 10 iterations. This signs the stability of the moving manipulator.

D. Path Tracking on Two Planes

Besides, when stability is essential in a task, we can make a trade-off by setting more targets along the path and sacrifice in rapidity in order to minimize the error, and this raises requirement of path tracking. We conduct path tracking tests on x-z and y-z vertical planes, finishing the tracking of quadrilateral paths for 5 times. Due to the similarity of results, we only show the performance of test on x-z plane in following figures. Fig. 10 illustrates the path tracking process. The movement is mainly stable, with repetitive fluctuations. The manipulator skews upwards on the upper horizon when it moves to the midpoint and downwards on the lower horizon due to marginal effect.

Fig. 11 shows the positional and directional errors in path tracking task, where the manipulator shows small errors on z-axis and direction. It is obvious that fluctuations of error are repetitive, which means the difficulty of reaching different points on the path is not the same. The reason is that in different position the conformity of estimated Jacobian model towards the reality is different: where the estimated is less precise, the algorithm needs more iteration times to converge. In Fig. 12 as a snapshot, the arm bends while keeping a stable direction during the movement.

Besides, we test the performance of the feedback controller in several practical scenarios, as described in Fig. 13.

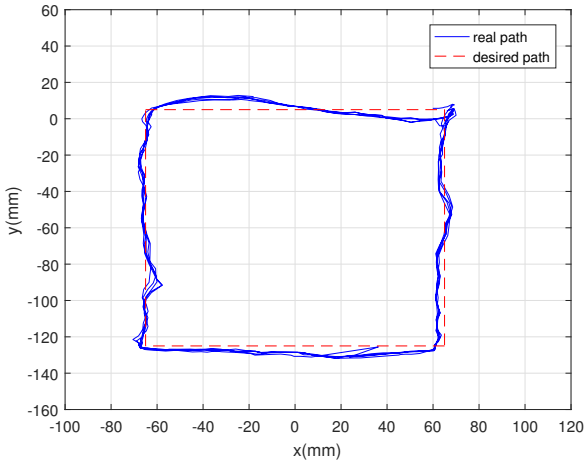

Fig. 10. Trajectory projection on the x-y plane of in path tracking task, where the dash line is the preset path. The clockwise cycle is repeated 5 times.

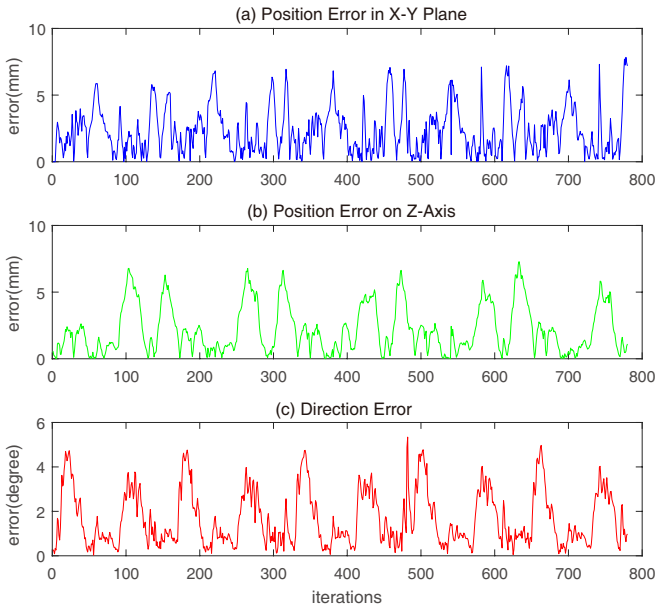

Fig. 11. Errors in path tracking task. Positional error in x-y plane, on z-axis, and orientational errors are respectively shown in (a), (b) and (c).

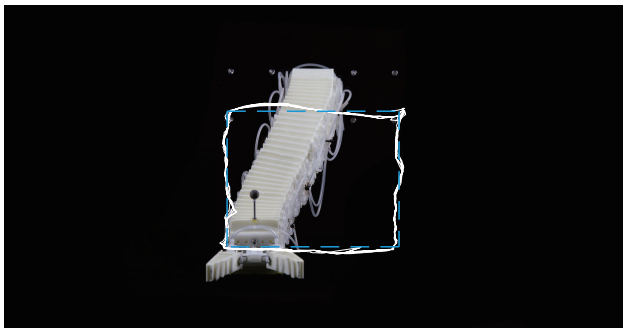

Fig. 12. A snapshot in path tracking task on the x-y plane. It can be figured out that even the arm curves, the direction of the end effector is stable.

IV. LIMITATION

The feedback control algorithm is developed on the basis of an estimated model, so its usable workspace will be restricted in where the model ensures convergence of iteration. Specifically, the restriction is that the angle between the end effector direction and z-axis cannot be more than 90°, beyond which the manipulator's extension will conversely cause coordinate reduction in z-axis. To solve this problem, we can elaborate the model by involving the positions of intermediate segments. The detailed model will contain information of the whole manipulator configuration and become more robust, thus, it will also be used in motion control. On the other side, the algorithm presented in this paper is more portable and operable if only the position and direction of the end effector are considered, because it needs less positional sensors and simpler geometrical model. Hysteresis effect is shown as an exceeding error about 10mm during the physical experiments, which is autonomously corrected thanks to the feedback mechanism, yet it should be examined independently. As for controller design, the proportional controller used in this work can be improved to more practical ones, PI, PID, etc. Besides, confined by the control system (current feedback frequency is 1Hz), the manipulator moves slowly in the path tracking task. In fact, the feedback frequency decides the overall performance of the current control system.

V. CONCLUSIONS

In this paper, we propose a position and direction control strategy of our manipulator based on estimated Jacobian-based model under uncertain load. The experimental results prove the claim that the combination of updating estimated model and feedback strategy is remarkable and highly applicable. In fact, the estimated model and feedback strategy are complementary. Specifically, as the model gets closer to a realistic model, a single-step will be more accurate, and the controlled movement can be executed more efficient. On the other side, when the model becomes less accurate, similar precision can still be achieved by employing feedback mechanism with sacrifice of rapidity. In future, we should further improve the feedback rate in order to improve the performance of our soft manipulator, which is a general development trend. As for the estimated model, it can be extended to involve whole manipulator's configuration in order to support more complex tasks, such as motion control. Besides, the proportional value can be supplanted by solenoids, which substantially reduces the cost and promote the popularization and applications of soft robots. 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.

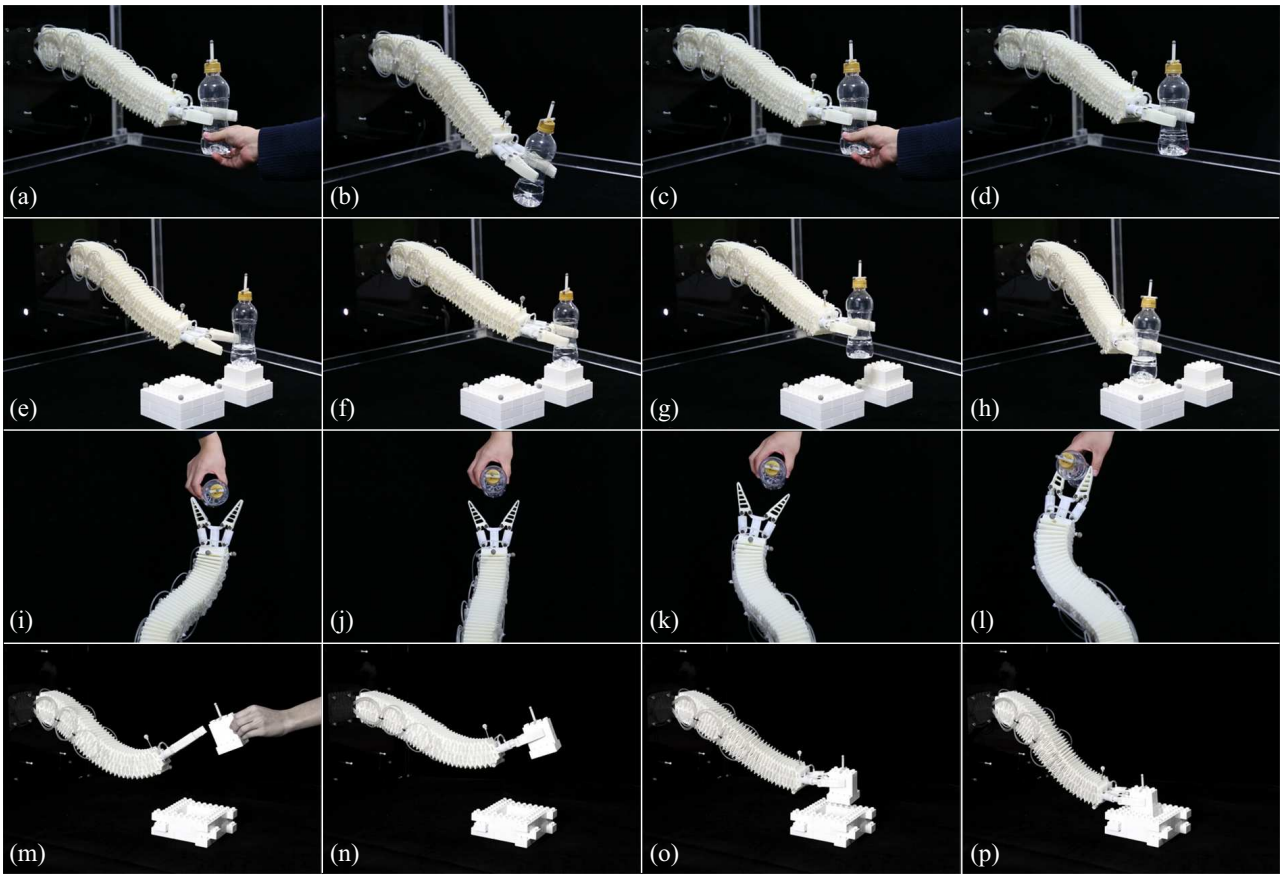

Fig. 13. This figure shows the process of finishing four tasks using HPN manipulator. In (a)-(d), the end effector falls when loaded with a 120g object, and then returns to the original position using feedback control. In (e)-(h), it moves the object from one position to another. In (i)-(l), it tracks the object's position and direction, shown from overlooking view. In (m)-(p), it grasps another cube-shaped object from a hand and place the object to the position.

[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. 20 400–20 403, 2011.

[5] T. Zheng, D. T. Branson, E. Guglielmino, and D. G. Caldwell, "A 3d dynamic model for continuum robots inspired by an octopus arm," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 3652–3657.

[6] 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.

[7] 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.

[8] H. Jiang, Z. Wang, X. Liu, X. Chen, Y. Jin, X. You, and X. Chen, "A two-level approach for solving the inverse kinematics of an extensible soft arm considering viscoelastic behavior," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 6127–6133.

[9] 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.

[10] H. Wang, W. Chen, X. Yu, T. Deng, X. Wang, and R. Pfeifer, "Visual servo control of cable-driven soft robotic manipulator," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 57–62.

[11] A. D. Marchese, K. Komorowski, C. D. Onal, and D. Rus, "Design and control of a soft and continuously deformable 2d robotic manipulation system," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2189–2196.

[12] P. Kormushev, Y. Demiris, and D. G. Caldwell, "Kinematic-free position control of a 2-dof planar robot arm," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5518–5525.

[13] M. C. Yip and D. B. Camarillo, "Model-less feedback control of continuum manipulators in constrained environments," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 880–889, 2014.

[14] M. C. Yip and D. B. Camarilo, "Model-less hybrid position/force control: a minimalist approach for continuum manipulators in unknown, constrained environments," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 844–851, 2016.

[15] 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 *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2016, pp. 350–356.