2024 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
July 15-19, 2024. Boston, MA, USA

Model Predictive Control of 2-DOF Tendon-Driven Continuum Robot Using Optical Tracking

Yilun Sun, *Member, IEEE*, Yuqing Liu, Ying Su, and Tim C. Lueth, *Senior Member, IEEE*

***Abstract*— In recent years, there has been a growing research focus on continuum robots due to their high flexibility and safety. Nevertheless, the inherent nonlinearity of the flexible structure of continuum robots has increased the complexity of their motion control. In this work, we propose a method based on model predictive control (MPC) to achieve the closed-loop motion control of a tendon-driven continuum robot. The robot has two bending degrees of freedom (DOFs) and the constant curvature model is used as the kinematic model. Selective laser sintering (SLS) technology is utilized to fabricate the entire continuum robot system, while a tracking camera is used to measure the robot position to provide the real-time feedback for the MPC controller. Experiments are also conducted, in which the continuum robot is actuated to move along different predefined plane trajectories. As a result, the position error of the MPC-based controller is much smaller than that of an open-loop controller, which demonstrates the good control performance of the proposed method.**

I. Introduction

Continuum robot is a type of robotic system composed of a flexible and deformable structure. Unlike conventional rigid-link robots, continuum robots are capable of achieving continuous and adaptable motions, allowing them to explore complex environments and perform tasks that require flexibility and dexterity. A typical application of the continuum robots is the minimally invasive surgery [1], as their flexible body can navigate through narrow and winding passages in the body, enabling procedures such as endoscopic surgeries and targeted interventions. Continuum robots also play an important role in search and rescue operations [2], where their flexibility allows them to navigate through rubble and debris to locate and aid trapped individuals. Furthermore, continuum robots have shown promise in the field of rehabilitation [3], assisting patients with mobility exercises and providing therapeutic interventions.

Due to the high flexibility of their continuum body, it is not easy to model and control continuum robots. Currently, the most commonly used kinematic model is the constant-curvature model [4], which simplifies the analysis and control of the robot's motion by approximating the bending shape of a continuum robot as a series of constant-curvature segments. Other high-fidelity modeling methods, such as the finite element method (FEM) [5], [6] or some deep learning-based methods [7], can improve the modeling accuracy of continuum robots, but they also increase the computational cost for motion control. As for control strategies, the proportional-integral-derivative (PID) controller was widely used in both open-loop and closed-loop control of continuum robots [8]. In order to further enhance the ability of continuum robots to interact with external environments, some researchers also proposed adaptive controllers [9], [10]. On the other hand, different sensors such as optical camera [9], magnetic sensor [11] and fiber bragg gratings (FBG) [12], were often used in the literature to provide shape estimation of continuum robots for closed-loop control.

The authors are with the Institute of Micro Technology and Medical Device Technology, Technical University of Munich, Garching, Germany (e-mail of the corresponding author: yilun.sun@tum.de).

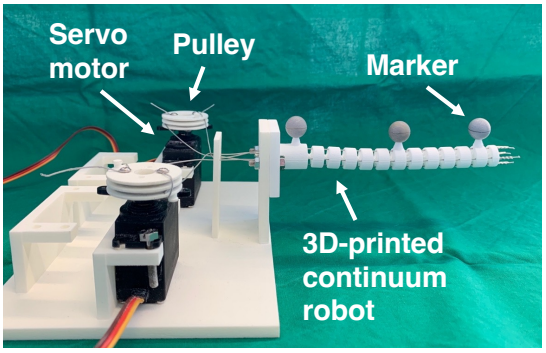

Fig. 1. A 3D-printed tendon-driven continuum robot with two segments.

In this work, we propose a model predictive control (MPC) based closed-loop approach to improve the control performance of continuum robots. An optical camera is utilized to generate position feedback. The MPC approach was chosen because it can also take into account the future changes of the desired motion trajectory when making control actions [13]. In addition, a 3D-printed two-segment continuum robot (see Fig. 1) was used to evaluate the performance of the proposed control method.

The rest of this paper is organized as follows. Section II describes the kinematic model of the two-segment continuum robot. The MPC-based controller is illustrated in Section III. Section IV presents the results of the control experiments. Conclusion is drawn in Section V.

II. Kinematic Model of the Continuum Robot

A. Geometry Design of the Robot

As is shown in Fig. 1, the tendon-driven continuum robot consists of two bending segments, which include two bending degrees of freedom (DOFs) in total. The entire continuum robot is constructed as a monolithic structure and a series of bio-inpired flexure joints [14], [15] are integrated to realize the flexible bending movements. In this work, we used the Solid Geometry (SG) Library [16], [17] in MATLAB for geometry modeling and the created 3D model of the continuum robot is presented in Fig. 2. Here, four cable

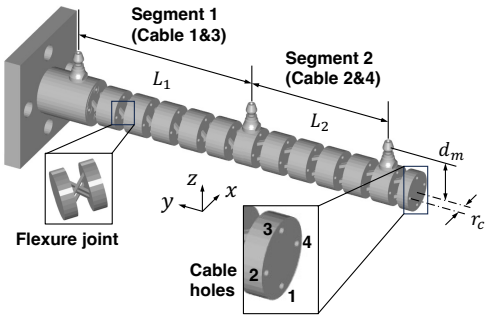

Fig. 2. 3D geometry model of the continuum robot in this work.

TABLE I

VALUES OF THE GEOMETRY PARAMETERS FOR THE CONTINUUM ROBOT

Parameter	Symbol	Value
Length of segment 1	L_1	58.5 mm
Length of segment 2	L_2	43.5 mm
Distance between cable hole and disk center	r_c	4.5 mm
Distance between disk center and marker	d_m	15 mm
Radius of the pulley on the servo motor	r_p	12.5 mm

holes are constructed in each rigid disk along the continuum backbone, so that two pairs of cables can be routed for the robot actuation. As shown in Fig. 1, the cable 1 and 3 (pair 1) are attached to the circular pulley of a servo motor to realize the bending motion of segment 1, while the segment 2 is actuated by cable 2 and 4 (pair 2). L_1 and L_2 are the length of the bending segment 1 and 2, respectively. r_c indicates the distance between the cable hole and the disk center. The radius of the pulley is denoted by r_p. In addition, the distance between the disk center and the position tracking marker is denoted by d_m. Table I shows the values of the geometry parameters.

B. Constant-Curvature-Based Kinematic Model

To formulate the forward kinematics of the created 2-DOF continuum robot, we should first define the parameters of its actuator space, configuration space, and task space.

Since we utilize two servo motors to actuate the continuum robot, the rotation angle ϕ_1 and ϕ_2 of the motors are chosen to represent the actuator space $\mathbf{q}$:

$$\mathbf{q} = \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} \tag{1}$$

The resulted length changes $\{\Delta l_i\}_{i=1,\dots,4}$ of the four cables can be formulated as:

$$\begin{pmatrix} \Delta l_1 \\ \Delta l_2 \\ \Delta l_3 \\ \Delta l_4 \end{pmatrix} = r_p \cdot \begin{pmatrix} \phi_1 \\ \phi_2 \\ -\phi_1 \\ -\phi_2 \end{pmatrix} \tag{2}$$

Based on the constant-curvature principle, the two bending segments can be described as two arcs with the curvature k_1 and k_2 (see Fig. 3 and Fig. 4), which are selected to represent the configuration space $\mathbf{k} = \begin{pmatrix} k_1 & k_2 \end{pmatrix}^T$. Here, we denote the bending angles of the two segments as θ_1 and θ_2. $\mathbf{p_1}$ and $\mathbf{p_2}$ represent the coordinates of marker 2 and 3. In the Cartesian

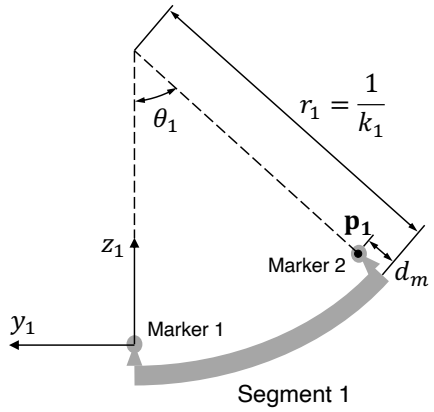

Fig. 3. Schematic diagram of the kinematic model of bending segment 1.

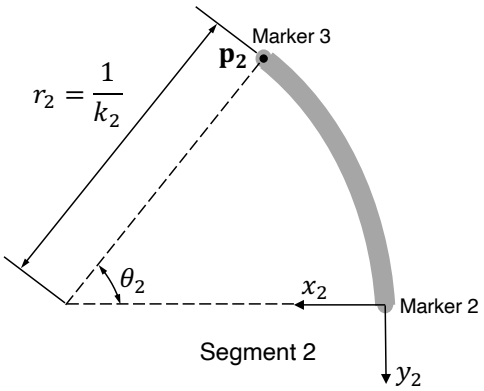

Fig. 4. Schematic diagram of the kinematic model of bending segment 2.

coordinate system of $x_1y_1z_1$ (as in Fig. 3), the relationship between the radius and length of the arc can be formulated as:

$$\theta_1 = \frac{r_1 - r_c}{L_1 - \Delta l_1} = \frac{r_1}{L_1} \implies \frac{r_c}{r_1} = \frac{\Delta l_1}{L_1} \tag{3}$$

where $r_1 = \frac{1}{k_1}$ is the bending radius of segment 1. Substituting (2) into (3) yields to:

$$k_1 = \frac{1}{r_1} = \frac{r_p \phi_1}{r_c L_1} \tag{4}$$

For the segment 2 in the coordinate system $x_2y_2z_2$ in Fig. 4, a similar relationship as (4) can be obtained:

$$k_2 = \frac{1}{r_2} = \frac{r_p \phi_2}{r_c L_2} \tag{5}$$

As is shown in Fig. 3 and Fig. 4, the local coordinates of marker 1 and 2 in the coordinate system $x_1y_1z_1$ and $x_2y_2z_2$ ($_\mathbf{1}\mathbf{p_1}$ and $_\mathbf{2}\mathbf{p_2}$) can be written as:

$$_\mathbf{1}\mathbf{p_1} = \begin{pmatrix} 0 \\ -(\frac{1}{k_1} - d_1) \cdot \sin\theta_1 \\ (\frac{1}{k_1} - d_1) \cdot (1 - \cos\theta_1) \end{pmatrix} \tag{6}$$

$$_\mathbf{2}\mathbf{p_2} = \begin{pmatrix} \frac{1}{k_2} \cdot (1 - \cos\theta_2) \\ -\frac{1}{k_2} \cdot \sin\theta_2 \\ 0 \end{pmatrix} \tag{7}$$

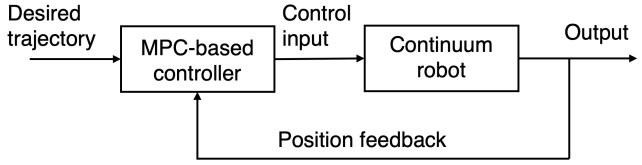

Fig. 5. Schematic diagram of the MPC-based control process.

The rotation matrix ${}_1\mathbf{A_2}$ between the two coordinate systems can be calculated based on the rotation angle θ_1:

$$ {}_1\mathbf{A_2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & \sin\theta_1 \\ 0 & -\sin\theta_1 & \cos\theta_1 \end{pmatrix} \tag{8} $$

In this work, we take $x_1y_1z_1$ as the global coordinate system. According to the chain rule, the global coordinate of marker 3 (${}_1\mathbf{p_2}$) can be calculated by using the following equation:

$$ {}_1\mathbf{p_2} = {}_1\mathbf{p_1} + {}_1\mathbf{A_2} \cdot {}_2\mathbf{p_2} \tag{9} $$

Here, we define ${}_1\mathbf{p_2}$ as the task space $\mathbf{f}$, which can be obtained by substituting (6), (7), (8) into (9):

$$ \mathbf{f} = \begin{pmatrix} \frac{1}{k_2} \cdot (1 - \cos\theta_2) \\ -(\frac{1}{k_1} - d_1) \cdot \sin\theta_1 - \frac{1}{k_2} \cdot \sin\theta_2 \cdot \cos\theta_1 \\ (\frac{1}{k_1} - d_1) \cdot (1 - \cos\theta_1) + \frac{1}{k_2} \cdot \sin\theta_2 \cdot \sin\theta_1 \end{pmatrix} \tag{10} $$

where $\theta_1 = k_1 L_1$ and $\theta_2 = k_2 L_2$.

By calculating the first-order derivative of $\mathbf{f}$ with respect to $\mathbf{q}$, the Jacobian matrix $\mathbf{J}$ of the forward kinematics can be calculated as:

$$ \mathbf{J}(\mathbf{q}) = \frac{\partial \mathbf{f}}{\partial \mathbf{k}} \cdot \frac{\partial \mathbf{k}}{\partial \mathbf{q}} \tag{11} $$

which describes the relationship between the end effector velocity $\dot{\mathbf{f}}$ and the angular velocity $\dot{\mathbf{q}}$ of the motor as:

$$ \dot{\mathbf{f}} = \mathbf{J}(\mathbf{q}) \cdot \dot{\mathbf{q}} \tag{12} $$

III. MPC-Based Control Strategy

The basic concept of our MPC-based control strategy is to use an optimization-based algorithm to determine the control input for the 2-DOF continuum robot, which takes into account not only the current position measured by the tracking camera, but also the future positions predicted by using the kinematic model from Section II-B. Fig. 5 shows the workflow of the proposed control strategy.

In this work, the continuum robot is supposed to complete the desired trajectory using t seconds with N_t steps. In each time step, a control input shall be generated by the MPC controller. Here, we denote $\mathbf{u}_i$ as the control input (2×1 vector) at the i-th step. The update of the actuator space can be expressed as:

$$ \mathbf{q}_i = \mathbf{q}_{i-1} + \mathbf{u}_i \tag{13} $$

where $\mathbf{q}_i$ and $\mathbf{q}_{i+1}$ represent the motor angles at the i-th and $(i+1)$-th step, respectively. With the help of (12), the end effector position at the $(i+k)$-th step can be approximated by using the measured i-th-step position $\mathbf{f}_{m,i}$ and the control sequence $\{\mathbf{u}_{i+j-1}\}_{j=1,\dots,k}$:

$$ \tilde{\mathbf{f}}_{i+k} \approx \mathbf{f}_{m,i} + \sum\nolimits_{j=1}^{k} \mathbf{J}(\mathbf{q}_{i+j-1}) \cdot \mathbf{u}_{i+j-1} \tag{14} $$

where $\tilde{\mathbf{f}}_{i+k}$ represents the predicted $(i+k)$-th-step position. In order to obtain the optimal control input for the i-th step ($\mathbf{u}_i$), we have defined the following optimization problem:

$$ \left.\begin{aligned} \min_{\mathbf{u}_i,\dots,\mathbf{u}_{i+N-1}} : C = \sum_{k=0}^{N-1} [Q \cdot (\tilde{\mathbf{f}}_{i+k} - \mathbf{f}_{i+k})^T (\tilde{\mathbf{f}}_{i+k} - \mathbf{f}_{i+k}) \\ + R \cdot \mathbf{u}_{i+k}^T \cdot \mathbf{u}_{i+k}] \\ \text{s.t.} : \mathbf{u}_{\min} \le \mathbf{u}_{i+j-1} \le \mathbf{u}_{\max}, \forall j \in \{1, \dots, N\} \end{aligned} \right\} \tag{15} $$

where C is the cost function and $\mathbf{f}_{i+k}$ is the $(i+k)$-th-step position of the desired trajectory. The upper and lower bound of the control input are denoted by $\mathbf{u}_{\max}$ and $\mathbf{u}_{\min}$, respectively. Herein, the first part of C describes the deviation between the desired trajectory and the real end effector position over the N prediction steps, while the second part depicts the control efforts for the N steps. Q and R are two positive weighting factors for balancing the two parts of C.

Here, by introducing the N prediction steps, the predicted future positions $\{\tilde{\mathbf{f}}_{i+k}\}_{k=1,\dots,N-1}$ are taken into account when computing the control input. It should be mentioned that, although a series of control input increments $\{\mathbf{u}_i, \dots, \mathbf{u}_{i+N-1}\}$ are calculated as the solution of the optimization problem (15), only the first increment $\mathbf{u}_i$ is taken as the control input for the i-th time step. Algorithm 1 shows the workflow of the proposed MPC-based control method, where the optimization problem is solved using the `fmincon` function from MATLAB (line 5).

Algorithm 1: MPC-Based Closed-Loop Control Algorithm

Input: Desired trajectory $\{\mathbf{f}_i\}_{i=1,\dots,N_t}$, Q, R, N, $\mathbf{u}_{\max}$, $\mathbf{u}_{\min}$
Output: Actuation sequence $\{\mathbf{q}_i\}_{i=1,\dots,N_t}$

1 $\mathbf{q}_1 \leftarrow \mathbf{f}^{-1}(\mathbf{f}_1)$;
2 Set the motor angles to $\mathbf{q}_1$;
3 **for** $i = 2 : N_t$ **do**
4 **if** $i + N - 1 > N_t$ **then**
5 $N \leftarrow N_t + 1 - i$;
6 **end**
7 Get the end effector position $\mathbf{f}_{m,i}$ from the camera;
8 Calculate C using (14) and (15);
9 Calculate the optimized $\{\mathbf{u}_i, ..., \mathbf{u}_{i+N-1}\}$ by solving (15);
10 $\mathbf{q}_i \leftarrow \mathbf{q}_{i-1} + \mathbf{u}_i$;
11 Update the motor angles using $\mathbf{q}_i$;
12 **end**

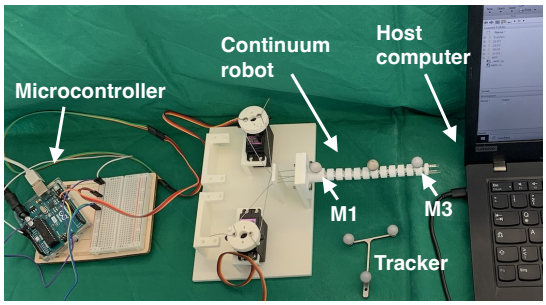

Fig. 6. Experimental setup for testing the MPC-based control method.

Algorithm 2: Open Loop Control Algorithm

Input: Desired trajectory $\{\mathbf{f}_i\}_{i=1,...,N_t}$
Output: Actuation sequence $\{\mathbf{q}_i\}_{i=1,...,N_t}$

$\mathbf{q}_1 \leftarrow \mathbf{f}^{-1}(\mathbf{f}_1)$;
Set the motor angles to $\mathbf{q}_1$;
for $i = 2 : N_t$ **do**
  $\mathbf{u}_i \leftarrow \mathbf{J}^{-1}(\mathbf{f}_{i-1}) \cdot (\mathbf{f}_i - \mathbf{f}_{i-1})$;
  $\mathbf{q}_i \leftarrow \mathbf{q}_{i-1} + \mathbf{u}_i$;
  Update the motor angles using $\mathbf{q}_i$;
end

IV. Experiments

A. Experimental Setup

Fig. 6 shows the experimental setup for evaluating the performance of the proposed control method. The continuum robot and its base were 3D-printed using selective laser sintering technology (FORMIGA P100, EOS GmbH, Germany), while two MG996R servo motors were used to drive the pulleys and tendons. An Arduino UNO microcontroller was chosen to actuate the motors. The end effector position of the continuum robot was measured by an optical tracking camera (Polaris Vicra, Northern Digital Inc., Canada), while the measured position difference between marker M3 and M1 was taken as $\mathbf{f}_{m,i}$. The microcontroller and the optical tracking camera were both connected to a host computer, where MATLAB was used to implement the proposed MPC method.

In this work, the weighting factors Q and R were empirically selected as 1 and 0.5, respectively, to achieve stable control performance. The prediction step number N was set to 4 to reduce the computational cost of the optimization problem. $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$ are set to $\begin{pmatrix}-2 & -2\end{pmatrix}^T$ and $\begin{pmatrix}2 & 2\end{pmatrix}^T$. During the experiments, the continuum robot was controlled to travel through ellipse, 8-shape and star-shape trajectories. In addition, we also compared the path tracking results of the proposed MPC method with an open-loop control method to evaluate its control performance. The algorithm of the open loop control method is presented in Algorithm 2, where $\mathbf{J}^{-1}$ is the inverse of the Jacobian matrix.

B. Path Tracking Results

1) Ellipse Trajectory: The first tracking path was an ellipse trajectory, whose y-axis and z-axis coordinates can be expressed as follows:

$$\begin{cases} y = y_{\max}\cos\left(\frac{\pi t}{40}\right), & t \in \{0, \Delta t, ..., (N_t-1)\Delta t\} \\ z = z_{\max}\sin\left(\frac{\pi t}{40}\right), & t \in \{0, \Delta t, ..., (N_t-1)\Delta t\} \end{cases} \quad (16)$$

where $y_{\max} = 35$mm and $z_{\max} = 45$mm are the maximum value of the ellipse trajectory on the y-axis and z-axis. $\Delta t = 1$s is the period of a control step and N_t is set to 80. Fig. 7 shows the desired ellipse trajectory in yz-plane and xz-plane, where the value of the x-axis coordinates is calculated by using (10).

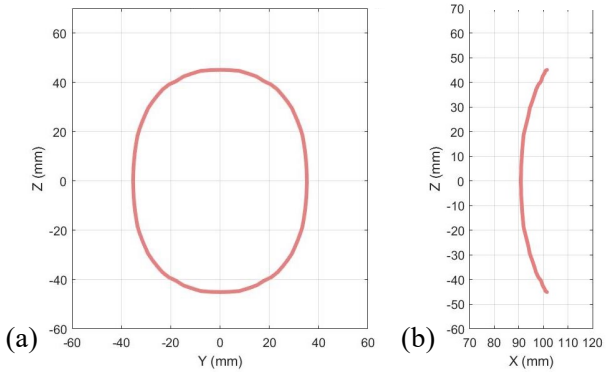

Fig. 7. Ellipse trajectory. (a) yz-plane. (b) xz-plane.

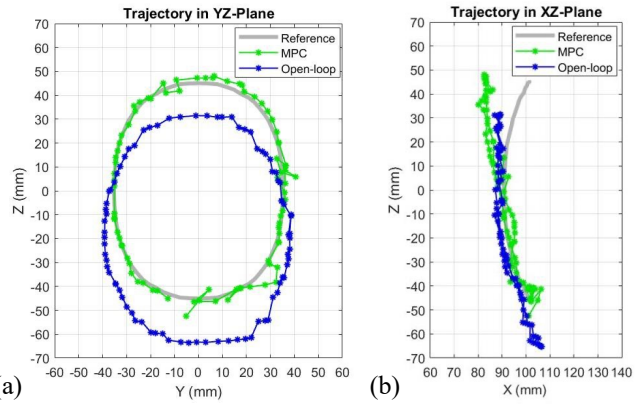

Fig. 8. Tracking test results for the ellipse trajectory. (a) yz-plane. (b) xz-plane.

Fig. 8 shows the tracking test results for the ellipse trajectory, where the green and blue curves represent the MPC-controlled and open-loop-controlled tracking paths, respectively. It can be seen that the MPC-controlled tracking path is generally consistent with the desired trajectory, while the open-loop-controlled path exhibits a shift (about 20mm) in the negative direction of the z-axis of the desired trajectory. The main reason for this phenomenon is that the gravity force of the continuum robot is not included in the constant curvature model, which leads to the z-axis position error of the open-loop-controlled path. By introducing the optical feedback and the proposed MPC algorithm, the z-axis error can be effectively eliminated. The position error diagram in Fig. 9 shows that the mean value of the error norm of the MPC-controlled path (7.5mm) is much smaller than that of the open-loop-controlled path (22.4mm), which also

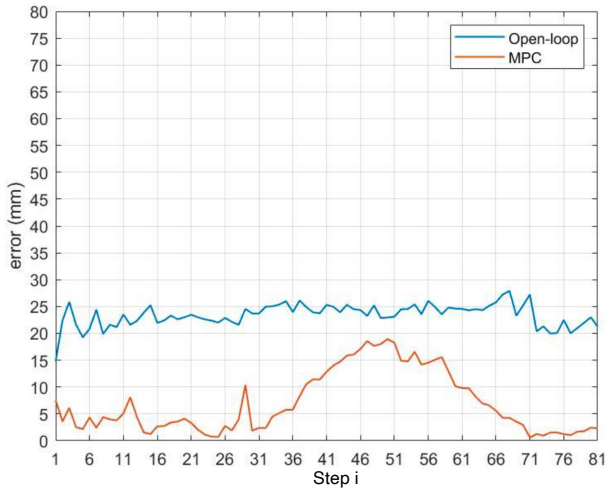

Fig. 9. Error norm of $|\mathbf{f}_{m,i} - \mathbf{f}_i|$ for the ellipse trajectory.

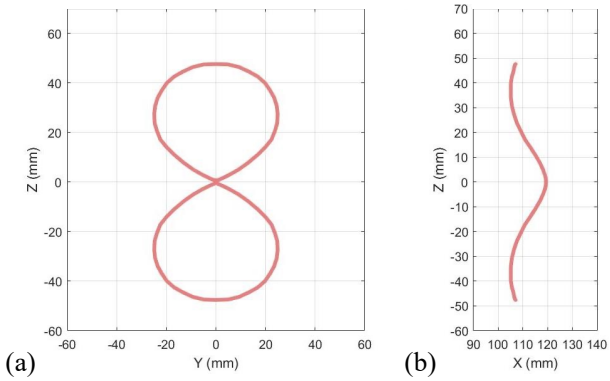

Fig. 10. 8-shape trajectory. (a) yz-plane. (b) xz-plane.

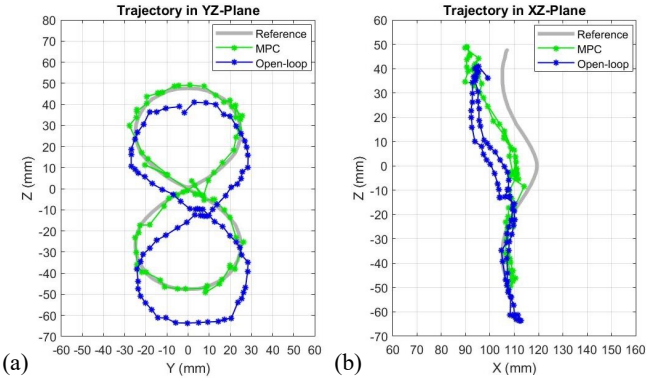

Fig. 11. Tracking test results for the 8-shape trajectory. (a) yz-plane. (b) xz-plane.

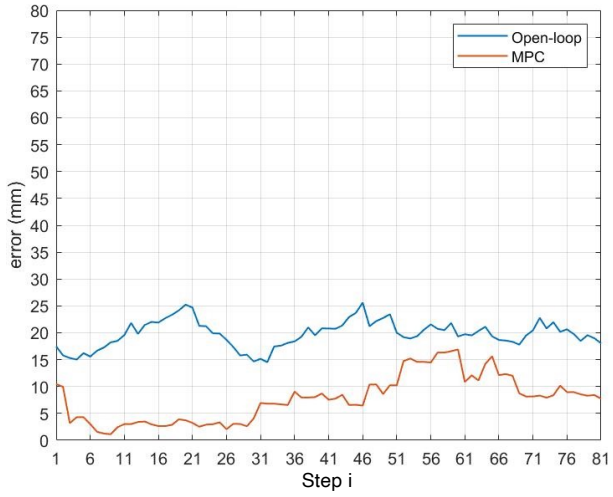

Fig. 12. Error norm of $|\mathbf{f}_{m,i} - \mathbf{f}_i|$ for the 8-shape trajectory.

verifies the good control performance of the proposed MPC algorithm.

2) 8-Shape Trajectory: The second tracking path was an 8-shape trajectory, whose y-axis and z-axis coordinates can be expressed as follows:

$$\begin{cases} y = z_{\max} \cdot \operatorname{sgn}\left(\cos\left(\frac{\pi t}{40}\right)\right) \cdot \sin\left(\frac{\pi t}{40}\right) \cdot \left(\cos\left(\frac{\pi t}{40}\right)\right)^2 \\ z = z_{\max} \cdot \operatorname{sgn}\left(\cos\left(\frac{\pi t}{40}\right)\right) \cdot \left(\cos\left(\frac{\pi t}{40}\right)\right)^2 \end{cases} \tag{17}$$

where $z_{\max} = 48$mm is the maximum value of the 8-shape trajectory on z-axis. The time range of t is the same as in the first case, which is $t \in \{0, \Delta t, ..., (N_t - 1)\Delta t\}$. Δt and N_t are set to 1s and 80, respectively. Fig. 10 shows the desired 8-shape trajectory in yz-plane and xz-plane.

Fig. 11 shows the tracking test results for the 8-shape trajectory. It can be noticed that, similar to the ellipse case, the MPC-controlled tracking path is also close to the desired trajectory, while the open-loop-controlled path exhibits a shift (about 15mm) in the negative direction of the z-axis of the desired trajectory. The position error diagram in Fig. 12 indicates that the mean value of the error norm of the MPC-controlled path (7.5mm) is much smaller than that of the open-loop-controlled path (21.3mm). Hence, for the 8-shape trajectory, the proposed MPC algorithm also shows better control performance than the open loop method.

3) Star-Shape Trajectory: The third tracking path was a star-shape trajectory, whose y-axis and z-axis coordinates can be expressed as follows:

$$\begin{cases} y = L_{\mathrm{s}} \cdot \left(\frac{2}{5} \cdot \sin\left(\frac{\pi t}{10}\right) - \sin\left(\frac{\pi t}{25}\right)\right) \\ z = L_{\mathrm{s}} \cdot \left(\frac{2}{5} \cdot \cos\left(\frac{\pi t}{10}\right) + \cos\left(\frac{\pi t}{25}\right)\right) \end{cases} \tag{18}$$

where L_{s} is set to 25mm. The time range of t is the same as in the first case, while Δt and N_t are set to 1s and 100, respectively. Fig. 13 shows the desired star-shape trajectory in yz-plane and xz-plane.

The tracking test results in Fig. 14 shows that, similar to the other two cases, the MPC-controlled tracking path is close to the desired trajectory, while the open-loop-controlled path exhibits a shift (about 15mm) in the negative direction of the z-axis of the desired trajectory. The position error diagram in Fig. 15 indicates that the mean value of the error norm of the MPC-controlled path (7mm) is much smaller than that of the open-loop-controlled path (18.5mm). Therefore, the control performance of the proposed MPC approach is also verified for the star-shape trajectory.

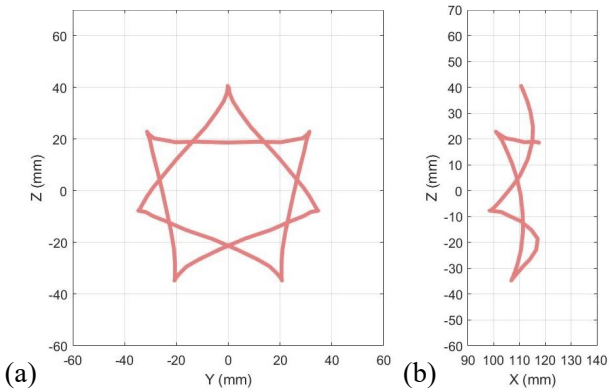

Fig. 13. Star-shape trajectory. (a) yz-plane. (b) xz-plane.

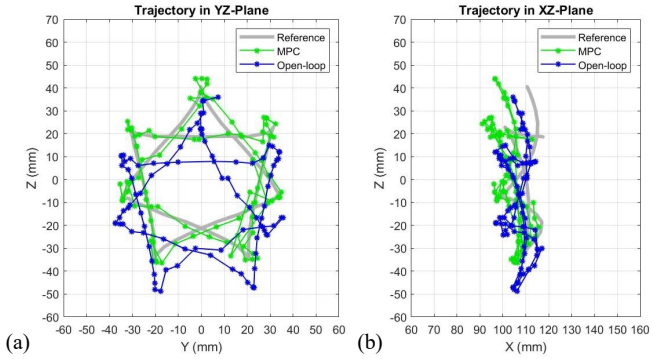

Fig. 14. Tracking test results for the star-shape trajectory. (a) yz-plane. (b) xz-plane.

V. Conclusion and Future Work

In this paper, we presented a MPC-based closed-loop method for controlling the motion of a 2-DOF tendon-driven continuum robot. The constant-curvature model was chosen for the kinematic modeling of the continuum robot, and a optical tracking camera was used to provide position feedback. Experimental results demonstrated that the proposed method can effectively reduce the position error of the continuum robot caused by external interference. In future work, we will use more complex tracking paths and external interference to further evaluate the feasibility of the proposed control method. On the other hand, advanced structural optimization [18], [19] will be performed on the continuum robot to improve the modeling accuracy of the kinematic model. In addition, we will further develop the MPC method to explore its applicability to controlling continuum robots with higher DOFs.

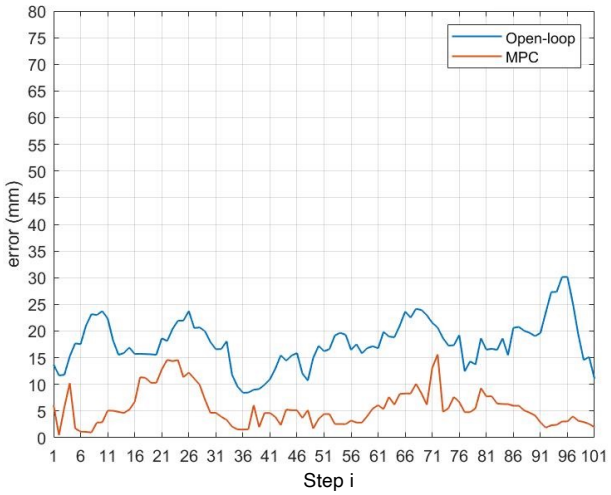

Fig. 15. Error norm of $|\mathbf{f}_{m,i} - \mathbf{f}_i|$ for the star-shape trajectory.

References

[1] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.

[2] X. Dong et al., "Continuum robots collaborate for safe manipulation of high-temperature flame to enable repairs in challenging environments," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 5, pp. 4217–4220, 2022.

[3] Z. Peng and J. Huang, "Soft rehabilitation and nursing-care robots: A review and future outlook," *Applied Sciences*, vol. 9, p. 3102, 2019.

[4] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[5] T. M. Bieze et al., "Finite element method-based kinematics and closed-loop control of soft, continuum manipulators," *Soft robotics*, vol. 5, no. 3, pp. 348–364, 2018.

[6] Y. Sun, D. Zhang, Y. Liu, and T. C. Lueth, "Fem-based mechanics modeling of bio-inspired compliant mechanisms for medical applications," *IEEE Trans. Med. Robot. Bionics*, vol. 2, no. 3, pp. 364–373, 2020.

[7] X. Wang, J. Dai, H.-S. Tong, K. Wang, G. Fang, X. Xie, Y.-H. Liu, K. W. S. Au, and K.-W. Kwok, "Learning-based visual-strain fusion for eye-in-hand continuum robot pose estimation and control," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2448–2467, 2023.

[8] J. Wang and A. Chortos, "Control strategies for soft robot systems," *Advanced Intelligent Systems*, vol. 4, no. 5, p. 2100165, 2022.

[9] H. Wang et al., "Visual servoing of soft robot manipulator in constrained environments with an adaptive controller," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 41–50, 2017.

[10] 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 Transactions on Mechatronics*, vol. 23, no. 1, pp. 286–297, 2018.

[11] S. Song, H. Ge, J. Wang, and M. Q.-H. Meng, "Real-time multi-object magnetic tracking for multi-arm continuum robots," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.

[12] S. Sefati et al., "Fbg-based control of a continuum manipulator interacting with obstacles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 6477–6483, IEEE, 2018.

[13] C. M. Best, M. T. Gillespie, P. Hyatt, L. Rupert, V. Sherrod, and M. D. Killpack, "A new soft robot control method: Using model predictive control for a pneumatically actuated humanoid," *IEEE Robotics & Automation Magazine*, vol. 23, no. 3, pp. 75–84, 2016.

[14] Y. Sun and T. C. Lueth, "Cruciate-ligament-inspired compliant joints: Application to 3d-printed continuum surgical robots," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pp. 4645–4648, 2021.

[15] Y. Sun and T. C. Lueth, "Safe manipulation in robotic surgery using compliant constant-force mechanism," *IEEE Transactions on Medical Robotics and Bionics*, vol. 5, no. 3, pp. 486–495, 2023.

[16] Y. Sun and T. C. Lueth, "Sgcl: A b-rep-based geometry modeling language in matlab for designing 3d-printable medical robots," in *Proc. IEEE 17th Int. Conf. Automat. Sci. Eng.*, pp. 1388–1393, 2021.

[17] Y. Sun, Y. Liu, F. Pancheri, and T. C. Lueth, "Larg: A lightweight robotic gripper with 3-d topology optimized adaptive fingers," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 4, pp. 2026–2034, 2022.

[18] Y. Sun, Y. Liu, and T. C. Lueth, "Optimization of stress distribution in tendon-driven continuum robots using fish-tail-inspired method," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3380–3387, 2022.

[19] Y. Sun and T. C. Lueth, "Enhancing torsional stiffness of continuum robots using 3-d topology optimized flexure joints," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 4, pp. 1844–1852, 2023.