

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

Position and Orientation Tracking Control of a Cable-Driven Tensegrity Continuum Robot

Fei Li, *Member, IEEE*, Hao Yang, Guoying Gu, *Senior Member, IEEE*,
Yongqing Wang, and Haijun Peng, *Member, IEEE*

Abstract—Trajectory tracking control of flexible continuum robots is challenging due to their inherent compliance and highly nonlinearity. Many related works exclude the control of the end's orientation, i.e., only the end's position is considered. In this paper, a differential-algebraic equations (DAEs) model-based instantaneous optimal control (IOC) framework for the end's position and orientation cooperative tracking of a cable-driven tensegrity continuum robot (TCR) is developed. Based on the tensegrity concept, a TCR is designed first as the control object, which can achieve multimode deformations such as bending, scoliosis, contraction, and the S- or J-shape. Then, the actuation of cables is introduced as the system kinematic constraints from the view of multibody dynamics, so that a control-oriented model of the TCR can be built by DAEs. Subsequently, the original continuous trajectory tracking problem is approximated for a series of IOC problems at each discrete time slot. Finally, considering the constraints of control input saturation, a linear complementarity problem (LCP) was derived for solving these IOC problems. The method provides an easy-to-implement and unified framework for addressing the trajectory tracking control issues of cable-driven continuum robots, which can improve the control performance of the position-only tracking controllers and exploit the TCR's advantages to handle more application scenarios. The advanced performance and potential applications of the proposed controller have been evaluated via several numerical simulations and experiments on the TCR prototype.

Index Terms—Position and orientation tracking control, tensegrity continuum robot, DAEs model, cable-driven, input saturation

This paragraph of the first footnote will contain the date on which you submitted your paper for review, which is populated by IEEE. This work was supported in part by the National Science Foundation of China under Grant U2241263 and Grant 12202091, and in part by the Liaoning Provincial Natural Science Foundation Program under Grant 2023-BS-061. This article was recommended for publication by Associate Editor xxxx and Editor xxxx upon evaluation of the reviewers' comments. (*Corresponding author: Haijun Peng*).

Fei Li, Hao Yang and Haijun Peng are with the School of Mechanics and Aerospace Engineering, the State Key Laboratory of Structural Analysis, Optimization and CAE Software for Industrial Equipment, Dalian University of Technology, Dalian 116024, China (e-mail: lfei@dlut.edu.cn; yanghao123@mail.dlut.edu.cn; hjpeng@dlut.edu.cn).

Guoying Gu is with the State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: guoyoying@sjtu.edu.cn).

Yongqing Wang is with the School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: yqwang@dlut.edu.cn).

This article has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the authors.

Digital Object Identifier

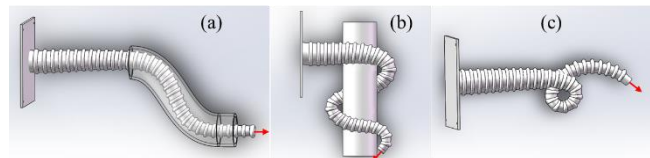


Fig. 1 Motion patterns of continuum robots: (a) insertion, (b) coiling, and (c) circumnutation

I. INTRODUCTION

A CONTINUUM robot [1]-[4] is a biologically inspired slender hyper-redundant mechanism, which provides remarkable adaptability to unstructured and difficult-to-access environments owing to its intrinsic compliance. In recent years, numerous scenario applications for continuum robots have been suggested, including in-situ inspection and repair for aeroengine [5], active hose for fluid delivery [6], operation in the nuclear industry [7], and minimally invasive surgeries with the da Vinci Surgical System. These applications usually require high-performance motion control strategies, which remains a thorny issue in continuum robots.

Actually, as presented in Fig. 1, the motion of continuum robots can be roughly classified into three typical patterns: insertion, coiling and circumnutation. Most existing motion control approaches of continuum robots essentially strive to tune these motion patterns to achieve certain required tasks. They can be divided according to their modeling methods: the model-based controllers [8], the model-free controllers based on some data-driven techniques such as machine learning and empirical methods [9], and the hybrid model controllers [10]. This research work is mainly focused on model-based control strategies. A popular motion control approach of continuum robots is to use arcs with constant curvature (CC) or piecewise constant curvature (PCC) [11] to roughly describe the body deformations; then certain kinematic model-based controllers can be designed. Similarly, some function parameterization models such as Bézier curve [12] can also be used to describe the deformation of continuum robots. But these kinematics models only based on the geometry of a robot can not properly describe the mechanics of its structure. Another modeling method for continuum robots is the pseudo-rigid-body model [13], which simplifies the continuum robot into a rigid chain interconnected by joints. However, this model may lack accuracy when addressing complex deformation patterns or rapidly moving objects. Therefore, researchers are beginning to seek control strategies based on dynamic models with considerations of internal and external forces. But the dynamic model-based control of continuum robots would be more complex, and is yet not well investigated, because of their

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

compliant nature and strong nonlinearity.

Herein, inspired by the fact that the end's Cartesian position and orientation are two crucial geometric features contained in the motion patterns, which can affect the overall configuration of a continuum robot, thus, we will focus on the trajectory tracking control for both the end's position and orientation of continuum robots. In the past few decades, various trajectory tracking control approaches for continuum robots have been developed, such as the proportional-derivative (PD) control [14], [15], the sliding mode control [16], [17] the model predictive control (MPC) [18], [19], and some other techniques [20], [21]. All these approaches have been proven to be effective for trajectory tracking control of continuum robots by numerical simulations or experiments. However, most of them ignore the end's orientation in favor of just controlling the end's position. Until recent years, a few works on the end's position and orientation control in the field of magnetic and pneumatic continuous robots have emerged [22]-[24]. But there are still few reports on this issue for the cable-driven tensegrity continuum robots.

In theory, the control performance of the aforementioned model-based trajectory tracking strategies can be significantly improved by implementing feedback on the robot prototype [25]; however, the introduction of the end's orientation will pose certain new challenges. First of all, it will increase the burden of feedback information measurement for a closed-loop controller. The motion control model of continuum robots is commonly built by the geometrically-derived PCC methods or the Cosserat rod theories [26], which contain attitude angles as basis variables in the governing equations. It usually needs to implant some specialized attitude measuring devices into the robot body to obtain the angles. Furthermore, the PCC or the Cosserat rod may not precisely reveal the configuration of the continuum robot, particularly for the proposed TCR with axial compression that will alter the body stiffness. Second, the solution to the control problem will be more complex. Mathematically, the tracking control of the end's orientation can be described as a nonlinear equality constraint concerning the state variables. When coupled with the large flexible deformation of the TCR, the entire control system will be a high-dimensional constrained nonlinear optimization problem, whose optimal solution is usually difficult to achieve.

In this work, based on the tensegrity structure [27], a special cable-driven continuum robot named the tensegrity continuum robot (TCR) is used as the control object; it will be introduced in the next section in detail. Then, to address the above two challenges on position and orientation tracking controls, we have made the following efforts accordingly.

For the former challenge, the end's orientation, actually, can be determined by the position coordinates of three non-collinear points on the end section. Thus, the measurement of the end's orientation can be replaced by that of point positions. Then, to better predict the elastic and large-scale deformations of the TCR, a positional finite element method (PFEM) was developed to establish the dynamic control model. Compared with the classical FEM [28], the absolute node positions in the global coordinate system are taken as the basic variables in the PFEM framework. A similar description can also be found in

[29], where a dynamic model of class-1 tensegrity is built based on the global position of the rod endpoints. As a result, the original sampling data of the robot's positions measured by a visual or other positioning sensor can be used as the feedback data without any further conversion. This can reduce the measuring errors and the sampling time to some extent. Furthermore, when the mass of sliding cables can be ignored, the system mass matrix by the PFEM will be constant [30]. So, it can avoid to calculate the additional inertia stiffness caused by the mass matrix when solving the Jacobian matrix of the dynamic system.

For the latter challenge, in order to control a continuum robot, the mapping between the low-level actuator space and the high-level task space that contains the end's position and orientation of the robot needs to be formulated first [31]. In fact, from the view of multibody dynamics, the actuation of cables for the TCR can be introduced as the system driving constraints, which can be described by kinematic constrained algebraic equations. Therefore, joined with the dynamic differential equations as stated above, the trajectory tracking control system of the TCR can essentially be described by the differential-algebraic equations (DAEs). In this way, the actuation lengths or speeds of the driving cables, i.e., the sliding cables here, can be employed as the control inputs, so that the controller will be compatible with the operating modes of commercial motors directly. However, the control theory of a DAEs system will be more complex than that of ordinary differential equations (ODEs) systems, which are commonly used in the literature on model-based control for continuum robots [32]-[34]. The challenge mainly comes from the fact that the global optimal solution, which satisfies the DAEs coupled with other control conditions such as input saturation is difficult to obtain. Therefore, in this work, inspired by the idea of the instantaneous optimal control (IOC) method, which was first introduced for seismically excited structural vibration control [35], the original global optimal control problem in the whole continuous time domain will be transformed into an instantaneous suboptimal problem at each discrete time slot. Then, a DAEs model-based IOC approach is developed. Additionally, the input saturation is taken into consideration, i.e., the actuation lengths or speeds should be restricted within safe bounds. The IOC problem will be further transformed into a linear complementarity problem (LCP).

The main contributions of this article are summarized as follows:

- 1) A control-oriented formulation of the end's position and orientation tracking for the TCR is established. On the one hand, the governing equations are described by the node's positions in the global coordinate system so that the sensor feedback information can be used directly. On the other hand, the cable actuation is transformed into system kinematic constraints with driving lengths or speeds as the control inputs so that the controller will be more compatible with the operating modes of most commercial motors.

- 2) A DAEs model-based IOC algorithm framework for the end's position and orientation collaborative tracking control of the TCR with input saturation is proposed. The core idea of the proposed controller is to transform the original optimal control problem in the continuous time domain into an IOC

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

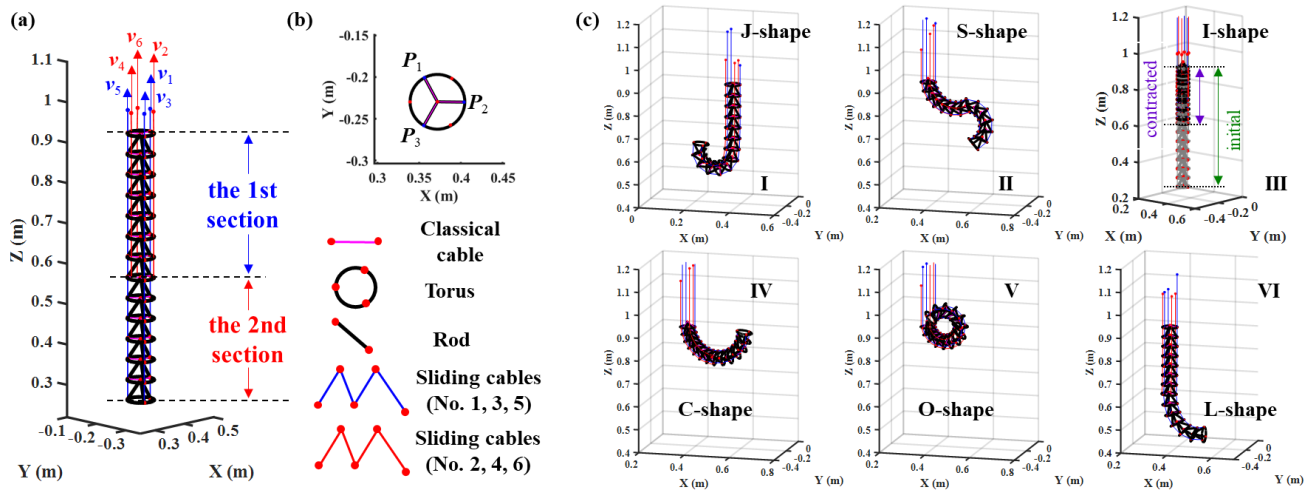


Fig. 2. Structural design and dexterous motion of the bio-inspired tensegrity continuum robot (TCR): (a)-(b) the designed robot consists of compression rods and torus, tension cables, and six sliding cable actuators; (c) a few configurations including the J-, S-, I-, C-, O- and L-shape generated by the proposed IOC controller.

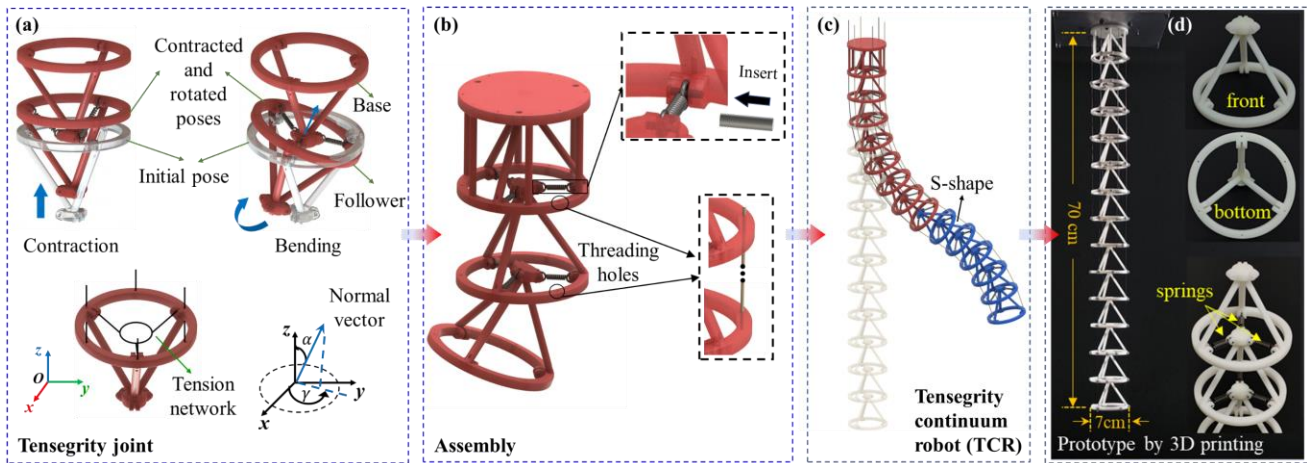


Fig. 3. Fabrication of the TCR: (a) the compliant tensegrity joint; (b) assembly by the tensegrity joints; (c) the body structure of the TCR; (d) the 3D printing experimental prototype.

problem at each discrete time slot so that a local suboptimal solution can be obtained. The effectiveness and potential applications of the proposed approach were illustrated by the experiments on the bio-inspired TCR.

The rest of this article is organized as follows. In Section II, the design and fabrication of a tensegrity continuum robot is presented. The dynamic model is derived briefly in Section III. Then, in Section IV, a DAEs model-based IOC approach for the end's position and orientation tracking control of the TCR is proposed in detail; and its stability is discussed. The control performance and the potential applications of the proposed controller are demonstrated by experiments and simulations in Section V. Finally, Section VI concludes this article.

II. STRUCTURAL DESIGN AND FABRICATION

Tensegrities [36]-[38] are a kind of self-stressed structures constructed with a set of compression elements suspended in a continuous network of tension cables, which have favorable properties such as high resistance-to-mass ratio and exhibiting dexterous motions with fewer actuators. Currently, despite

several continuum robots employing tensegrity have been developed [39]-[41], they are all single section with relatively simple motion modes. Here, as shown in Fig. 2(a), a spine-inspired tensegrity continuum robot (TCR) is designed as a compliant serial structure composed of several compression elements (rods and torus) and tension cables (or springs). The compression components imitate the vertebral bones, while the tension cables can realize the function of intervertebral discs and ligaments to help stabilize the overall structure. Driven by six sliding cables distributed around the periphery of the robot body, the proposed TCR can achieve multimode deformations, including the J-, S-, I-, C-, O-, and L-shape as presented in Fig. 2(c). These configurations were designed by the tracking controller proposed in this article without increasing the structural and operational complexity.

Specifically, Fig. 3 shows the fabrication processes of our TCR. Firstly, three triangularly distributed support rods are connected with a torus, which form a basic module of the TCR, i.e., a conical vertebra. Then, three tension cables are used to connect two adjacent modules (a base and a follower) and provide a tension network to support the follower module

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

suspended below the base module. This connection structure can actually be regarded as a tensegrity joint with six degrees of freedom (DOF). Compared to a rigid rotational joint, the flexible tensegrity joint not only has low friction without any direct contact between stiff components, but it also offers three small ranges of translational DOF, especially, the axial contraction motion DOF that can greatly expand the robot's workspace. Furthermore, the tensegrity joint is also a kind of variable-stiffness mechanism that can change stiffness by adjusting the prestress forces of the tension cables. Finally, as shown in Fig. 2(a), a two-segment TCR can be obtained by assembling all modules with the compliant tensegrity joints and configuring 6 sliding cable actuators on the periphery of the robot body in two groups. The sliding cables (blue lines) numbered 1, 3, and 5 pass through the entire robot body, while the other sliding cables (red lines) numbered 2, 4 and 6 only cross the holes from the base to the 8th torus. The 3D printing prototype is presented in Fig. 3(d). It consists of one base, 13 modules, 6 high-strength nylon cables, several tension springs and pins. The total length and cross-sectional diameter of the TCR are 0.7 m and 0.07 m, respectively.

Before the end of this section, some excellent features of the proposed TCR are summarized as follows. (a) Thanks to the compliant tensegrity joints, the TCR allows a wide range of axial contraction motion, which makes it more dexterous; (b) Benefiting from the tensegrity design concept, the TCR has a high volume-mass ratio and is very lightweight; (c) The structural stiffness of the TCR can be changed just by altering prestress of the tension components.

III. DYNAMIC MODELING AND PARAMETER IDENTIFICATION

From the perspective of structural dynamics, the FEM such as the complete Lagrange method [42] or the co-rotating coordinate method [43] can be used to establish the model of tensegrity structures with geometric nonlinearity. However, in the classical FEM, element formulation should be derived first in the local coordinate system; and then, be transformed to the global coordinate system. This transformation involves a large number of trigonometric function matrix operations, which is tedious and time-consuming. In order to overcome this defect, a so-called position FEM (PFEM) will be adopted for the TCR modeling in this work. Based on the PFEM, all the node position coordinates are taken as the generalized coordinates to describe the TCR's configuration; the dynamic equations can be established in the global coordinate system directly. So that the feedback, i.e., the current configuration of the TCR can be measured directly by visual equipment, without any data conversion or analysis. To this end, the data reliability and sampling efficiency of the real-time feedback for the subsequent close-loop controls can both be guaranteed. The dynamic modeling and system parameter identification are presented in the following subsections.

A. Dynamic Modeling Based on the PFEM

As shown in Fig. 4, the TCR is made up of four types of elements: rod, classical cable, sliding cable and torus. To avoid ambiguity, three modeling assumptions of the TCR have been noted: 1) the rods are slender so that the rotation effects

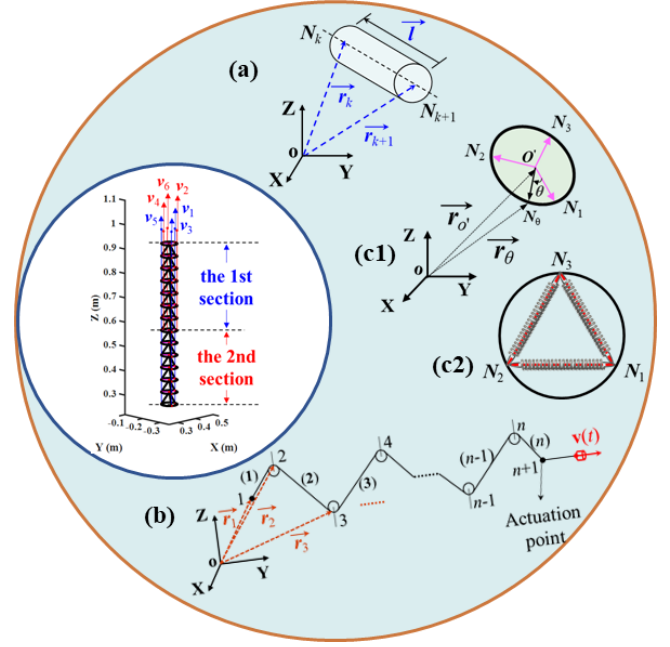


Fig. 4. Elements of the TCR: (a) a rod or classical cable element, (b) a sliding cable element, (c1-c2) a perspective and top view of the torus element, respectively.

along the axial direction can be neglected; and their bucking behavior are not considered; 2) the friction at the contact point between two elements is ignored. Based on these assumptions, the dynamic model of the TCR can be built using the PFEM. The detailed derivation of dynamic formulas for the first three elements can be seen in our previous work [30]. For the torus element, the generalized coordinates and the matrices of mass, stiffness and damping are derived as follows.

1) *Derivations for the Torus Elements:* Firstly, as shown in Fig. 4(c1), the torus element can be described by the three vertices (N_1, N_2, N_3) of its inscribed regular triangle. The generalized coordinates of the torus element are given as

$$\mathbf{q}_s = [\mathbf{r}_1^T \ \mathbf{r}_2^T \ \mathbf{r}_3^T]^T = [x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3]^T \quad (1)$$

where \mathbf{r}_i denotes the position coordinates of node N_i . To obtain the position coordinates for any point named N_θ on the torus element, the angle θ between $\overline{o'N_i}$ and $\overline{o'N_\theta}$ is introduced. Let $\mathbf{r}_\theta = \overline{oN_\theta}$, $\mathbf{r}_{o'} = \overline{o'o'}$, $\mathbf{n}_i = \overline{o'N_i}$ ($i = 1, 2, 3$), the position coordinates of a point N_θ can be derived as

$$\mathbf{r}_\theta = \mathbf{r}_{o'} + \overline{o'N_\theta} = \mathbf{r}_{o'} + \frac{2}{3} \left[\cos(\theta) \cdot \mathbf{n}_1 + \cos\left(\theta + \frac{2\pi}{3}\right) \cdot \mathbf{n}_2 + \cos\left(\theta + \frac{4\pi}{3}\right) \cdot \mathbf{n}_3 \right] \quad (2)$$

where $\mathbf{n}_i = \mathbf{r}_i - \mathbf{r}_{o'}$, $\mathbf{r}_{o'} = (\mathbf{r}_1 + \mathbf{r}_2 + \mathbf{r}_3) / 3$, $i = 1, 2, 3$.

Subsequently, we assume that: (a) the thickness, i.e., the cross-section area of the torus element is small enough; and (b) the rotational effect can be ignored. Based on these two assumptions, the kinetic energy can be derived, given as

$$T_s = \int_0^{2\pi} 0.5 \rho b \dot{\mathbf{r}}_\theta^T \dot{\mathbf{r}}_\theta d\theta = 0.5 \dot{\mathbf{q}}_s^T \left(\int_0^{2\pi} \rho b \mathbf{A}_\theta^T \mathbf{A}_\theta d\theta \right) \dot{\mathbf{q}}_s \quad (3)$$

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

where $A_\theta = \partial \mathbf{r}_\theta / \partial \mathbf{q}_s \in \mathbb{R}^{3 \times 9}$, b and ρ are the radius and the linear density of the torus element, respectively. Thereby, the mass matrix of the torus element can be obtained, given as

$$\mathbf{M}_s = \int_0^{2\pi} \rho b A_\theta^T A_\theta d\theta = \frac{2\pi}{3} \rho b \mathbf{I}_9 \quad (4)$$

Then, according to the stress characteristics of the torus elements, as shown in Fig. 4(c2), the generalized internal force of the torus element \mathbf{f}_s can be approximately simulated by the superimposed force of three virtual rod elements \mathbf{f}_{ri} (i.e., the sides of the inscribed triangle) on the nodes N_1 , N_2 and N_3 . The equivalent stiffness matrix and damping matrix of the torus element, i.e., \mathbf{K}_s and \mathbf{C}_s , can be obtained just by superposing and assembling the stiffness and damping of the above three virtual rod elements, given as

$$\mathbf{f}_s = \bigcup_{i=1,2,3} \mathbf{f}_{ri}, \quad \mathbf{K}_s = \bigcup_{i=1,2,3} \mathbf{K}_{ri}, \quad \mathbf{C}_s = \bigcup_{i=1,2,3} \mathbf{C}_{ri} \quad (5)$$

where the operator $\bigcup(*)$ denotes the finite element assembly, the specific expressions of \mathbf{K}_r and \mathbf{C}_r can be found in [30].

Furthermore, the boundary conditions of the cable-driven actuation should be introduced.

2) *Boundary Conditions of the Cable-Driven Actuation:* In this work, inspired by the fact that many physical actuation modes for the continuum robots such as the gas-driven mode, the liquid-driven mode and the cable-driven mode can be directly transformed into geometric changes. That is, the actuation boundary conditions can be translated into the corresponding geometric constraints. For example, the cable-driven mode can be described as the length constraints, and the fluid-driven mode including gas-driven and liquid-driven can be described as the area or volume constraints. Based on this idea, the boundary conditions of the sliding cable-driven actuation for the TCR will be constructed.

As shown in Fig. 4(b), the sliding cable can be connected to an actuator at the endpoint N_{n+1} to drive the TCR for bending, contracting, and other movements. The real displacement of the actuation point N_{n+1} should be equal to the length driven by the actuator. Thus, the kinematic boundary condition Φ_i of the i^{th} sliding cable can be obtained.

$$\Phi_i(\mathbf{q}, t) = d_{ci}^{\text{end}}(\mathbf{q}_{ci}, t) + \int_0^t v_i(t) dt \quad (6)$$

where \mathbf{q}_{ci} denotes the generalized coordinates of the i^{th} sliding cable, v_i denotes the actuation speed by the actuator, d_{ci}^{end} denotes the driving length, which can be calculated by the rest length l'_{ci} , given as

$$d_{ci}^{\text{end}}(\mathbf{q}_{ci}, t) = l'_{ci}(\mathbf{q}_{ci}(t)) - l'_{ci}(\mathbf{q}_{ci}(0)), \quad \text{and} \quad l'_{ci} = \sum_{k=1}^{n-1} l_{ci}^{(k)} \quad (7)$$

where $l_{ci}^{(k)}$ denotes the length of the k^{th} section ($\overline{N_{k+1}N_k}$) of the sliding cable. If the point N_n of the sliding cable is fixed and the actuation point N_{n+1} moves along the straight line $L_{N_n, N_{n+1}}$, d_{ci}^{end} can be simplified as

$$d_{ci}^{\text{end}}(\mathbf{q}_{ci}, t) = \|\mathbf{q}_{ci}^{\text{end}}(t) - \mathbf{q}_{ci}^{\text{end}}(0)\| \quad (8)$$

where $\mathbf{q}_{ci}^{\text{end}}$ denotes the position coordinates of the actuation point for the i^{th} sliding cable. Then, all kinematic boundary

conditions, which belong to the holonomic constraints, can be assembled, i.e., $\Phi(\mathbf{q}, t) = [\Phi_1, \Phi_2, \dots, \Phi_m]^T = \mathbf{0}$, where m is the number of active sliding cables.

Finally, the Lagrange function of the TCR system can be derived by assembling the mass matrix and the generalized forces of all elements. Based on the second Lagrange equation, the dynamic model of the cable-driven TCR can be achieved.

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} + \Phi_q^T \lambda + \mathbf{f}_{in}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{f}_{ext} = \mathbf{0} \\ \Phi(\mathbf{q}, t) = \mathbf{0} \end{cases} \quad (9)$$

where (9) is known as the differential-algebraic equations (DAEs), which contains the differential equation (the 1st one) and the algebraic constraint equation (the 2nd one). $\mathbf{M} \in \mathbb{R}^{n \times n}$ denotes the system mass matrix; $\lambda \in \mathbb{R}^{m \times 1}$ denotes the Lagrange multipliers; n and m are the number of generalized degree of freedom and holonomic constraints, respectively.

The item $\Phi_q^T \lambda$ physically represents the driving force vector applied to the active sliding cables. $\mathbf{f}_{in} \in \mathbb{R}^{n \times 1}$ denotes the generalized internal force vector including the elastic forces and damping forces; and $\mathbf{f}_{ext} \in \mathbb{R}^{n \times 1}$ denotes the external force vector.

Remark 1: It notes that \mathbf{M} will be a constant matrix when the mass of the sliding cables is small enough to ignore their dynamic effects. The detail discussions can be found in [30].

B. Parameter Identification Using the PSO Approach

To make the theoretical model (9) can more accurately estimate the deformation and other mechanical behaviors of the TCR, some unknown but critical parameters such as elastic modulus need to be identified. According to the essential idea of the parameter identification technology, the identification problem can be described in a way that finds the optimal parameters to minimize the mean position errors between simulation and experiment under certain constraints. It can be formulated as a constrained optimization problem.

$$\begin{cases} \text{find } \zeta \\ \min_{\zeta} . p = \frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \|\bar{\mathbf{q}}_{ij} - \hat{\mathbf{q}}_{ij}\| \\ \text{s.t. Eq. (9) with unknown parameters } \zeta \\ \zeta_{\min.} \leq \zeta \leq \zeta_{\max.} \end{cases} \quad (10)$$

where ζ is the model parameter to be identified; $\zeta_{\min.}$ and $\zeta_{\max.}$ are the corresponding feasible boundary values. $\bar{\mathbf{q}}_{ij}$ and $\hat{\mathbf{q}}_{ij}$ are the node position coordinates obtained by the theoretical model-based simulations and the experimental measurements with the motion capture system, respectively. The subscript 'i' and 'j' represent the results obtained at the j^{th} moment ($j=1, 2, \dots, n_2$) with the i^{th} group actuation inputs ($i=1, 2, \dots, n_1$).

In this work, as shown in Fig. 3, the TCR prototype is a rigid-flexible system. Because the rod and torus elements are relatively rigid, their elastic modulus deviation has little impact on the mechanical responses; while the classical cables (the springs) and the driving sliding cables (the nylon ropes) are flexible components, their elastic modulus will be the main factors for determining the system's inherent stiffness. Thus, only the elastic modulus of the classical cables and the sliding cables are chosen for material parameters identification; while

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

other parameters are given based on their theoretical values.

After determining the model parameters to be identified and achieving the experimental sampling data, a mature intelligent optimization algorithm, named particle swarm optimization (PSO) will be used to solve the above optimization problem (10). The PSO solver can be called in MATLAB software by a script function “particleswarm”.

IV. POSITION AND ORIENTATION TRACKING CONTROL WITH INPUT SATURATION

In this section, based on the DAEs model, an IOC approach for the position and orientation tracking control of the TCR with input saturation is proposed. The formula derivation and stability discussion can be found in the following subsections.

A. Problem Formulation

Driving by several sliding cables, the position and orientation collaborative tracking control of the TCR can be described as a kind of nonlinear optimal control problem (NOCP):

$$\left\{ \begin{array}{l} \text{find } \mathbf{u}(t) \\ \text{min. } J = \int_0^{t_f} \left[\begin{array}{l} (\mathbf{y} - \tilde{\mathbf{y}}(t))^T \cdot \hat{\mathbf{Q}}_1 \cdot (\mathbf{y} - \tilde{\mathbf{y}}(t)) + \\ (\mathbf{n} - \tilde{\mathbf{n}}(t))^T \cdot \hat{\mathbf{Q}}_2 \cdot (\mathbf{n} - \tilde{\mathbf{n}}(t)) + \\ \mathbf{u}^T(t) \cdot \hat{\mathbf{R}} \cdot \mathbf{u}(t) \end{array} \right] dt \\ \text{s.t. } \mathbf{M}\ddot{\mathbf{q}} + \Phi_q^T \lambda - \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0} \\ \Phi(\mathbf{q}, \mathbf{u}, t) = \mathbf{0} \\ \mathbf{y} = h_1(\mathbf{q}, t) = \frac{1}{3} \sum_{k=1}^3 \mathbf{r}_k \\ \mathbf{n} = h_2(\mathbf{q}, t) = \mathbf{d} / \|\mathbf{d}\|, \mathbf{d} = (\mathbf{r}_3 - \mathbf{r}_2) \otimes (\mathbf{r}_2 - \mathbf{r}_1) \\ \mathbf{u}_{\min} \leq \mathbf{u}(t) \leq \mathbf{u}_{\max} \end{array} \right. \quad (11)$$

where t_f is terminal times; $\hat{\mathbf{Q}}_k \in \mathbb{R}^{3 \times 3}$ ($k=1, 2$) and $\hat{\mathbf{R}} \in \mathbb{R}^{3 \times 3}$ are the nonnegative and positive definite symmetric weighting matrices, respectively. h_1 and h_2 are the output functions for computing the end's position $\mathbf{y}(\mathbf{q}, t) \in \mathbb{R}^{3 \times 1}$ and its orientation $\mathbf{n}(\mathbf{q}, t) \in \mathbb{R}^{3 \times 1}$, respectively; the operators ‘ \otimes ’ and ‘ $\|\cdot\|$ ’ denote the cross product and the Euclidean distance, respectively; $\tilde{\mathbf{y}}(t) \in \mathbb{R}^{3 \times 1}$ and $\tilde{\mathbf{n}}(t) \in \mathbb{R}^{3 \times 1}$ are the target position and the target orientation; $\mathbf{F} = \mathbf{f}_{\text{ext}} - \mathbf{f}_{\text{in}}$ is the apparent force vector; the control inputs $\mathbf{u}(t) \in \mathbb{R}^{m \times 1}$ (the driving speeds or lengths) is not free but subject to the box constraints of input saturation shown in (11). So far, the DAEs model-based tracking control problem of both the end position and orientation for the TCR has been formulated.

The NOCP (11) is an open-loop control problem for the DAEs system, which aims to find the optimal solution in the entire continuous time domain $[0, t_f]$. Existing strategies for solving this problem can be roughly categorized as direct or indirect methods. The direct methods transform the original control problem into a nonlinear programming (NLP) problem by a discrete scheme [44]. They are favored for their intuitive and unified solution framework to different control systems and constraints. However, due to the high-dimensional nature of TCR, the scale of such an NLP problem will be very large.

It will take much time or maybe fail to achieve convergence. The indirect methods transform the original control problem into a series of nonlinear equations by deriving the first-order optimality necessary conditions in state space [45], [46]. These methods ensure at least local optimality but require a reasonable initial guess, which can be challenging to provide.

In this work, to overcome the above difficulties in solving the NOCP, inspired by the idea of the IOC approach, the original global optimal control problem in the continuous time domain $[0, t_f]$ can be discretized into a series of suboptimal control problems at each time step. Thus, the computational burden of the controller can be significantly reduced. Besides, since the IOC is a close-loop controller running step by step, the results of the last step can serve as the initial values for the current step. Therefore, the difficulty of initial guessing can naturally be avoided. In the following subsections, the IOC approach for the position and orientation tracking control of the TCR will be presented in detail.

B. Algorithm Construction of the IOC controller

To establish the IOC controller, the continuous time domain $[0, T]$ is discretized into N time slots with equal time step length η , $\eta = T / N$. Then, the dynamic model of the TCR, i.e., the equality constraints DAEs in (11) should be discretized by some numerical integration schemes. Here, an implementation using the generalized- α scheme is presented.

In the generalized- α algorithm, the basic assumptions for the generalized coordinates \mathbf{q} and velocities $\dot{\mathbf{q}}$ in the discrete-time slot $[t_k, t_{k+1}]$ are as follows

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \eta \dot{\mathbf{q}}_k + \eta^2 (0.5 - \beta) \mathbf{a}_k + \eta^2 \beta \mathbf{a}_{k+1} \quad (12)$$

$$\dot{\mathbf{q}}_{k+1} = \dot{\mathbf{q}}_k + \eta (1 - \gamma) \mathbf{a}_k + \eta \gamma \mathbf{a}_{k+1} \quad (13)$$

where \mathbf{a} is an acceleration-like auxiliary variable. It is defined by the recurrence relation.

$$\left\{ \begin{array}{l} (1 - \alpha_m) \mathbf{a}_{k+1} + \alpha_m \mathbf{a}_k = (1 - \alpha_f) \ddot{\mathbf{q}}_{k+1} + \alpha_f \ddot{\mathbf{q}}_k \\ \mathbf{a}_0 = \ddot{\mathbf{q}}_0 \end{array} \right. \quad (14)$$

where the parameters α_m , α_f , β and γ can be selected to have suitable accuracy and stability properties. Herein, a selection method developed by Chung et. al [47] is adopted. That is,

$$\left\{ \begin{array}{l} \alpha_m = (2\rho - 1) / (\rho + 1), \quad \alpha_f = \rho / (\rho + 1) \\ \beta = (1 + \alpha_f - \alpha_m)^2 / 4, \quad \gamma = 0.5 + \alpha_f - \alpha_m \end{array} \right. \quad (15)$$

where $\rho \in [0, 1]$ denotes the spectral radius of the algorithm, which is a measure of numerical dissipation; a smaller ρ corresponds to greater numerical dissipation. In this work, ρ is set to 0.8. In addition, the variables \mathbf{q} , $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ should satisfy the DAEs at time t_{k+1} . Thus, we have

$$\bar{\mathbf{f}}_1 = \mathbf{M}\ddot{\mathbf{q}}_{k+1} + \Phi_q^T(\mathbf{q}_{k+1})\lambda_{k+1} - \mathbf{F}(\mathbf{q}_{k+1}, \dot{\mathbf{q}}_{k+1}, t_{k+1}) = \mathbf{0} \quad (16)$$

$$\bar{\mathbf{f}}_2 = \Phi(\mathbf{q}_{k+1}, \mathbf{u}_{k+1}, t_{k+1}) = \mathbf{0} \quad (17)$$

With analytical handling of (12)-(14), $\dot{\mathbf{q}}_{k+1}$, $\ddot{\mathbf{q}}_{k+1}$ and \mathbf{a}_{k+1} can be expressed in terms of the unknown \mathbf{q}_{k+1} . Thereby, the DAEs in (11) now is discretized into nonlinear algebraic equations (16) and (17) with respect to \mathbf{q}_{k+1} , λ_{k+1} and \mathbf{u}_{k+1} .

Herein, taking into account (6) and (8), the specific form of (17) can be derived as follows. On the one hand, if the control

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

inputs \mathbf{u}_{k+1} are the driving lengths of the sliding cables during each time step $[t_k, t_{k+1}]$, (17) can be expressed as

$$\Phi_i(\mathbf{q}_{k+1}) = \|\mathbf{q}_{ci}^{end}(t_{k+1}) - \mathbf{q}_{ci}^{end}(t_k)\| + \mathbf{u}_{k+1}^i \quad (18)$$

Furthermore, when the actuation point moves along a certain coordinate axis, such as the Z-direction, (18) can be further simplified to

$$\Phi_i(\mathbf{q}_{k+1}) = \mathbf{q}_{zi}^{end}(t_{k+1}) - \mathbf{q}_{zi}^{end}(t_k) + \mathbf{u}_{k+1}^i \quad (19)$$

where \mathbf{q}_{zi}^{end} denotes the displacement of the actuation point for the i^{th} sliding cable. On the other hand, if the control inputs \mathbf{u}_{k+1} are the driving speeds, it is reasonable to assume that the sliding cables are driven at a constant speed during each time step. Under this assumption, (17) can be expressed as

$$\Phi_i(\mathbf{q}_{k+1}) = \mathbf{q}_{zi}^{end}(t_{k+1}) - \mathbf{q}_{zi}^{end}(t_k) + \mathbf{u}_{k+1}^i \cdot \eta \quad (20)$$

In this work, the driving speeds are used as control inputs for the controller design. However, due to the servo motors operate in position control mode, the corresponding actuation lengths, $\eta\mathbf{u}_{k+1}$, are applied to the TCR.

Then, if \mathbf{u}_{k+1} is considered a known quantity, the remaining unknowns \mathbf{q}_{k+1} and λ_{k+1} in (16) and (17) can be solved using the Newton-Raphson iterative scheme. Specifically, let

$$\mathbf{W}(\mathbf{z}_{k+1}) = [\bar{\mathbf{f}}_1^T, \bar{\mathbf{f}}_2^T]^T = \mathbf{0} \quad (21)$$

where $\mathbf{z}_{k+1} = (\mathbf{q}_{k+1}^T, \lambda_{k+1}^T)^T$. Taking into account (20), the above (21) can be rewritten as

$$\mathbf{W}(\mathbf{z}_{k+1}) = \hat{\mathbf{f}}(\mathbf{z}_{k+1}) + \mathbf{B}\mathbf{u}_{k+1} = \mathbf{0} \quad (22)$$

where

$$\hat{\mathbf{f}}(\mathbf{z}_{k+1}) = \begin{Bmatrix} \bar{\mathbf{f}}_1(\mathbf{z}_{k+1}) \\ \hat{\mathbf{G}}(\mathbf{q}_{k+1}) \end{Bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_{n \times m} \\ \eta\mathbf{I}_m \end{bmatrix} \quad (23)$$

$$\hat{\mathbf{G}}(\mathbf{q}_{k+1}) = \begin{Bmatrix} \mathbf{q}_{z1}^{end}(t_{k+1}) - \mathbf{q}_{z1}^{end}(t_k) \\ \mathbf{q}_{z2}^{end}(t_{k+1}) - \mathbf{q}_{z2}^{end}(t_k) \\ \vdots \\ \mathbf{q}_{zm}^{end}(t_{k+1}) - \mathbf{q}_{zm}^{end}(t_k) \end{Bmatrix}, \quad \mathbf{u}_{k+1} = \begin{Bmatrix} \mathbf{u}_{k+1}^1 \\ \mathbf{u}_{k+1}^2 \\ \vdots \\ \mathbf{u}_{k+1}^m \end{Bmatrix} \quad (24)$$

Using the Newton-Raphson iterative scheme, the iterative equation for corrections \mathbf{z}_{k+1} can be derived as

$$\begin{aligned} \mathbf{z}_{k+1}^{(j+1)} &= \mathbf{z}_{k+1}^{(j)} - \mathbf{W}'_{\mathbf{z}_{k+1}}{}^{-1}(\mathbf{z}_{k+1}^{(j)}) \cdot \mathbf{W}(\mathbf{z}_{k+1}^{(j)}) \\ &= \mathbf{z}_{k+1}^{(j)} - \hat{\mathbf{f}}'_{\mathbf{z}_{k+1}}{}^{-1}(\mathbf{z}_{k+1}^{(j)}) \cdot \hat{\mathbf{f}}(\mathbf{z}_{k+1}^{(j)}) - \hat{\mathbf{f}}'_{\mathbf{z}_{k+1}}{}^{-1}(\mathbf{z}_{k+1}^{(j)}) \cdot \mathbf{B} \cdot \mathbf{u}_{k+1} \end{aligned} \quad (25)$$

where the superscript indicates the iteration index, $\mathbf{z}_{k+1}^{(j+1)}$ is the variable of the current $(j+1)^{\text{th}}$ iteration at time t_{k+1} , and $\mathbf{z}_{k+1}^{(j)}$ is the variable from the last j^{th} iteration that is taken as the reference for the current computation. Herein, to accelerate convergence, the results at the previous time step are used as the initial value for the iterations at the current time step, i.e., $\mathbf{z}_{k+1}^{(0)} = \mathbf{z}_k$. In addition, the Jacobian matrix $\hat{\mathbf{f}}'_{\mathbf{z}_{k+1}}$ can be given as

$$\hat{\mathbf{f}}'_{\mathbf{z}_{k+1}} = \begin{bmatrix} \mathbf{M}\hat{\boldsymbol{\beta}} + (\Phi_q^T \lambda)_q - \mathbf{F}_q - \mathbf{F}_q \hat{\boldsymbol{\gamma}} & \Phi_q^T \\ \Phi_q & \mathbf{0} \end{bmatrix} \mathbf{q} = \mathbf{q}_{k+1}^{(j)}, \lambda = \lambda_{k+1}^{(j)} \quad (26)$$

where

$$\hat{\boldsymbol{\beta}} = \partial \ddot{\mathbf{q}} / \partial \mathbf{q} = \mathbf{I} / \beta \eta^2, \quad \hat{\boldsymbol{\gamma}} = \partial \dot{\mathbf{q}} / \partial \mathbf{q} = \gamma \mathbf{I} / \beta \eta \quad (27)$$

It is worth noting that the Jacobian matrix (26) may be poorly conditioned. Therefore, an optimal preconditioning scheme for DAEs reported in [48] is used to improve the condition

number in this work. The details are not shown here again. To simplify the presentation, (25) can be rewritten as

$$\mathbf{z}_{k+1}^{(j+1)} = \boldsymbol{\xi}_1^{(j)} - \boldsymbol{\xi}_2^{(j)} \cdot \mathbf{u}_{k+1} \quad (28)$$

where

$$\boldsymbol{\xi}_1^{(j)} = \mathbf{z}_{k+1}^{(j)} - \hat{\mathbf{f}}'_{\mathbf{z}_{k+1}}{}^{-1}(\mathbf{z}_{k+1}^{(j)}) \cdot \hat{\mathbf{f}}(\mathbf{z}_{k+1}^{(j)}), \quad \boldsymbol{\xi}_2^{(j)} = \hat{\mathbf{f}}'_{\mathbf{z}_{k+1}}{}^{-1}(\mathbf{z}_{k+1}^{(j)}) \cdot \mathbf{B} \quad (29)$$

So far, the unknown variable at the current iteration step $\mathbf{z}_{k+1}^{(j+1)}$ is obtained, which is an explicit expression with respect to the control input vector \mathbf{u}_{k+1} .

Finally, the cost function J in (11) should be redefined. In the traditional IOC framework, the control cost function J is usually a weighted quadratic function of the errors and inputs at each time grid t_k ($0 \leq k \leq N$). In this work, the errors contain the central position error and the orientation error at the end of the TCR. Because the end torus element approximates a rigid body, the end's position \mathbf{y} and orientation \mathbf{n} can be described by the end's three non-collinear points, or separately described by one center point and two rotation angles θ_x and θ_y , as defined in Section V(D). In this work, for the following two reasons, the former was used to describe the end's position and orientation of the TCR. Firstly, in the PFEM framework, the basic unknown variables are the absolute node's position coordinates \mathbf{q} in the global coordinate system. These variables can intuitively and uniformly capture the information about robot's configuration and motion state, without the need to introduce additional angle variables. On the contrary, introducing attitude angles will increase system's nonlinearity, since they should be expressed as nonlinear functions of \mathbf{q} as shown in (11). As a result, the cost function J will also be nonlinear, rather than quadratic. It will make the problem more complex and even fail to solve. Secondly, when a visual motion capture system is adopted for measurements, the end's node positions captured by the visual system can be directly used as the feedback data without any conversion or analysis. However, the end's orientation angles θ_x and θ_y can not be measured directly by the visual system, which need to be obtained using the measured \mathbf{q} to solve a nonlinear equation online. This will not only affect the accuracy of measurement, but also increase the computational time cost of the controller.

Thus, the original position and orientation collaborative tracking control task of the TCR can be equivalently converted to the position tracking control for only the three vertices (P_1, P_2, P_3) of the inscribed regular triangle, as shown in Fig. 2(b). In addition, to alleviate the oscillation problem, the increment of control inputs can be introduced into the cost function J . Then, based on the aforementioned iterative scheme and target transformation, the IOC algorithm for the end's position and orientation tracking control of the TCR can be constructed.

$$\begin{cases} \text{find } \mathbf{u}_{k+1} \\ \text{min. } \hat{J} = (\hat{\mathbf{y}}_{k+1} - \tilde{\mathbf{y}}_{d,k+1})^T \cdot \hat{\mathbf{Q}} \cdot (\hat{\mathbf{y}}_{k+1} - \tilde{\mathbf{y}}_{d,k+1}) \\ \quad + \mathbf{u}_{k+1}^T \cdot \hat{\mathbf{R}} \cdot \mathbf{u}_{k+1} + \Delta \mathbf{u}_{k+1}^T \cdot \hat{\mathbf{W}} \cdot \Delta \mathbf{u}_{k+1} \\ \text{s.t. } \mathbf{z}_{k+1}^{(j+1)} = \boldsymbol{\xi}_1^{(j)} - \boldsymbol{\xi}_2^{(j)} \mathbf{u}_{k+1} \\ \quad \mathbf{u}_{\min} \leq \mathbf{u}_{k+1} \leq \mathbf{u}_{\max} \end{cases} \quad (30)$$

where \hat{J} is the modified cost function; $\hat{\mathbf{Q}} \in \mathbb{R}^{9 \times 9}$, $\hat{\mathbf{R}} \in \mathbb{R}^{m \times m}$ and $\hat{\mathbf{W}} \in \mathbb{R}^{m \times m}$ are the control weighting matrices; $\tilde{\mathbf{y}}_{d,k+1} \in \mathbb{R}^{9 \times 1}$

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

is the target vector, which contains the desired end position and orientation information; $\Delta \mathbf{u}_{k+1} = \mathbf{u}_{k+1} - \mathbf{u}_k$ is the increment of the control inputs; $\hat{\mathbf{y}}_{k+1} \in \mathbb{R}^{9 \times 1}$ denotes the output vector, i.e., the nine position coordinates of the end's three nodes, which can be calculated by the following output equation

$$\hat{\mathbf{y}}_{k+1} = \hat{\mathbf{C}} \cdot \mathbf{z}_{k+1}^{(j+1)} = \hat{\xi}_1 - \hat{\xi}_2 \mathbf{u}_{k+1} \quad (31)$$

where $\hat{\mathbf{C}} \in \mathbb{R}^{9 \times (n+m)}$ is the output matrix; $\hat{\xi}_i = \hat{\mathbf{C}} \xi_i^{(j)}$ ($i = 1, 2$).

C. Solution Based on the LCP Theory

To find the suboptimal control inputs \mathbf{u}_{k+1} , the IOC shown in (30), actually a constrained optimization problem, will be transformed into an unconstrained optimization problem first, and further transformed into an LCP.

For the equality constraints, it can be addressed just by substituting (31) into the modified cost function in (30).

$$\begin{aligned} \hat{J} = & (\hat{\xi}_1 - \hat{\xi}_2 \mathbf{u}_{k+1} - \tilde{\mathbf{y}}_{d,k+1})^T \cdot \hat{\mathbf{Q}} \cdot (\hat{\xi}_1 - \hat{\xi}_2 \mathbf{u}_{k+1} - \tilde{\mathbf{y}}_{d,k+1}) \\ & + \mathbf{u}_{k+1}^T \cdot \hat{\mathbf{R}} \cdot \mathbf{u}_{k+1} + \Delta \mathbf{u}_{k+1}^T \cdot \hat{\mathbf{W}} \cdot \Delta \mathbf{u}_{k+1} \end{aligned} \quad (32)$$

For the inequality constraints, i.e., the input saturation, they can be transformed into the following equality constraints by introducing two relaxation factors $\bar{\boldsymbol{\sigma}}_{k+1} \geq 0$ and $\underline{\boldsymbol{\sigma}}_{k+1} \geq 0$.

$$\begin{cases} -\mathbf{u}_{max} + \mathbf{u}_{k+1} + \bar{\boldsymbol{\sigma}}_{k+1} = \mathbf{0} \\ \mathbf{u}_{min} - \mathbf{u}_{k+1} + \underline{\boldsymbol{\sigma}}_{k+1} = \mathbf{0} \end{cases} \quad (33)$$

Then, based on the parametric variational principle [49], an expanded cost function can be defined as

$$\begin{aligned} \hat{J}_c = & (\hat{\xi}_1 - \hat{\xi}_2 \mathbf{u}_{k+1} - \tilde{\mathbf{y}}_{d,k+1})^T \hat{\mathbf{Q}} (\hat{\xi}_1 - \hat{\xi}_2 \mathbf{u}_{k+1} - \tilde{\mathbf{y}}_{d,k+1}) \\ & + \mathbf{u}_{k+1}^T \hat{\mathbf{R}} \mathbf{u}_{k+1} + \Delta \mathbf{u}_{k+1}^T \hat{\mathbf{W}} \Delta \mathbf{u}_{k+1} \\ & + \bar{\boldsymbol{\kappa}}_{k+1}^T (\mathbf{u}_{k+1} - \mathbf{u}_{max} + \bar{\boldsymbol{\sigma}}_{k+1}) + \underline{\boldsymbol{\kappa}}_{k+1}^T (\mathbf{u}_{min} - \mathbf{u}_{k+1} + \underline{\boldsymbol{\sigma}}_{k+1}) \end{aligned} \quad (34)$$

where $\bar{\boldsymbol{\kappa}}_{k+1} \in \mathbb{R}^{m \times 1}$ and $\underline{\boldsymbol{\kappa}}_{k+1} \in \mathbb{R}^{m \times 1}$ are the parametric variables.

So far, the constrained optimization problem (30) has been changed into the unconstrained optimization problem (34).

Subsequently, to minimize the expanded cost function, the variation \hat{J}_c with respect to \mathbf{u}_{k+1} should be zero. That is,

$$\hat{\partial} \hat{J}_c / \partial \mathbf{u}_{k+1} = \mathbf{0} \quad (35)$$

Thereby, the control inputs \mathbf{u}_{k+1} can be obtained and explicitly expressed only on the parametric variables, given as

$$\mathbf{u}_{k+1} = -\hat{\mathbf{H}}_1 (\bar{\boldsymbol{\kappa}}_{k+1} - \underline{\boldsymbol{\kappa}}_{k+1} - 2\hat{\mathbf{W}}\mathbf{u}_k) + \hat{\mathbf{H}}_2 (\hat{\xi}_1 - \tilde{\mathbf{y}}_{d,k+1}) \quad (36)$$

where

$$\hat{\mathbf{H}}_1 = (\hat{\xi}_2^T \hat{\mathbf{Q}} \hat{\xi}_2 + \hat{\mathbf{R}} + \hat{\mathbf{W}})^{-1} / 2, \quad \hat{\mathbf{H}}_2 = 2\hat{\mathbf{H}}_1 \hat{\xi}_2^T \hat{\mathbf{Q}} \quad (37)$$

Then, substituting (36) into (33), yields

$$\begin{aligned} & \left\{ \begin{aligned} & \hat{\mathbf{H}}_2 (\hat{\xi}_1 - \tilde{\mathbf{y}}_{d,k+1}) - \mathbf{u}_{max} + 2\hat{\mathbf{H}}_1 \hat{\mathbf{W}} \mathbf{u}_k \\ & \mathbf{u}_{min} - \hat{\mathbf{H}}_2 (\hat{\xi}_1 - \tilde{\mathbf{y}}_{d,k+1}) - 2\hat{\mathbf{H}}_1 \hat{\mathbf{W}} \mathbf{u}_k \end{aligned} \right\} + \\ & \left[\begin{array}{cc} -\hat{\mathbf{H}}_1 & \hat{\mathbf{H}}_1 \\ \hat{\mathbf{H}}_1 & -\hat{\mathbf{H}}_1 \end{array} \right] \left\{ \begin{array}{c} \bar{\boldsymbol{\kappa}}_{k+1} \\ \underline{\boldsymbol{\kappa}}_{k+1} \end{array} \right\} + \left\{ \begin{array}{c} \bar{\boldsymbol{\sigma}}_{k+1} \\ \underline{\boldsymbol{\sigma}}_{k+1} \end{array} \right\} = \mathbf{0} \end{aligned} \quad (38)$$

Consequently, the constrained IOC problem (30) can finally be transformed into an LCP as follows.

$$\begin{cases} \hat{\boldsymbol{\phi}} + \hat{\mathbf{T}} \hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}} = \mathbf{0} \\ \hat{\boldsymbol{\alpha}} \geq \mathbf{0}, \hat{\boldsymbol{\beta}} \geq \mathbf{0}, \hat{\boldsymbol{\beta}}^T \hat{\boldsymbol{\alpha}} = \mathbf{0} \end{cases} \quad (39)$$

where

Algorithm 1: The IOC for Position and Orientation Control

Input: The target position $\tilde{\mathbf{y}}$, the target orientation $\tilde{\mathbf{n}}$ and the generalized- α algorithm parameter ρ .

Output: \mathbf{u}_{k+1} , \mathbf{q}_{k+1} , $\dot{\mathbf{q}}_{k+1}$, $\ddot{\mathbf{q}}_{k+1}$, $\boldsymbol{\lambda}_{k+1}$, ($k = 1, 2, \dots, N$)

- 1: Read model information and initial values, including \mathbf{M} , \mathbf{q}_0 , $\dot{\mathbf{q}}_0$, $\ddot{\mathbf{q}}_0$, $\boldsymbol{\lambda}_0$, \mathbf{u}_0
 - 2: Calculate equivalent targets $\tilde{\mathbf{y}}_d$ offline.
 - 3: Set weighting matrices $\hat{\mathbf{Q}}$, $\hat{\mathbf{R}}$, $\hat{\mathbf{W}}$, and convergence error ε .
 - 4: **for** time step $k = 0$ to N **do**
 - 5: $\mathbf{a}_{k+1} := (\alpha_f \ddot{\mathbf{q}}_k - \alpha_m \mathbf{a}_k) / (1 - \alpha_m)$
 - 6: $\mathbf{q}_{k+1}^{(0)}, \dot{\mathbf{q}}_{k+1}^{(0)} \leftarrow$ By Eqs. (12) and (13)
 - 7: $\ddot{\mathbf{q}}_{k+1}^{(0)} := \mathbf{0}$, $\boldsymbol{\lambda}_{k+1}^{(0)} := \boldsymbol{\lambda}_k$, $\mathbf{z}_{k+1}^{(0)} := \left\{ \left\{ \mathbf{q}_{k+1}^{(0)} \right\}^T, \left\{ \dot{\mathbf{q}}_{k+1}^{(0)} \right\}^T \right\}^T$
 - 8: Get current target positions $\tilde{\mathbf{y}}_{d,k+1}$
 - 9: Set $j = 0$
 - 10: **while** $\|\mathbf{z}_{k+1}^{(j+1)} - \mathbf{z}_{k+1}^{(j)}\| / \|\mathbf{z}_{k+1}^{(j+1)}\| > \varepsilon$ **do**
 - 11: Form and update the model matrices \mathbf{F} , $\boldsymbol{\Phi}$, $\boldsymbol{\Phi}_q$
 - 12: Update the function \hat{J} and its Jacobian matrix \hat{J}'_z
 - 13: $\xi_1^{(j)}, \xi_2^{(j)} \leftarrow$ By Eq. (29)
 - 14: $\bar{\boldsymbol{\kappa}}_{k+1}, \underline{\boldsymbol{\kappa}}_{k+1} \leftarrow$ Solving the LCP (39)
 - 15: $\mathbf{u}_{k+1}, \mathbf{z}_{k+1}^{(j+1)} \leftarrow$ By Eqs. (28) and (36)
 - 16: $\dot{\mathbf{q}}_{k+1}^{(j+1)} := \dot{\mathbf{q}}_{k+1}^{(j)} + \hat{\boldsymbol{\beta}} \cdot \Delta \mathbf{q}$, $\ddot{\mathbf{q}}_{k+1}^{(j+1)} := \ddot{\mathbf{q}}_{k+1}^{(j)} + \hat{\boldsymbol{\gamma}} \cdot \Delta \mathbf{q}$
 - 17: $j := j+1$
 - 18: **end while**
 - 19: $\mathbf{u}_k, \mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k, \boldsymbol{\lambda}_k \leftarrow \mathbf{u}_{k+1}, \mathbf{q}_{k+1}, \dot{\mathbf{q}}_{k+1}, \ddot{\mathbf{q}}_{k+1}, \boldsymbol{\lambda}_{k+1}$
 - 20: $\mathbf{a}_k := \mathbf{a}_{k+1} + (1 + \alpha_f) / (1 - \alpha_m) \cdot \dot{\mathbf{q}}_{k+1}$
 - 21: **end for**
-

$$\hat{\boldsymbol{\phi}} = \begin{cases} \hat{\mathbf{H}}_2 (\hat{\xi}_1 - \tilde{\mathbf{y}}_{d,k+1}) - \mathbf{u}_{max} + 2\hat{\mathbf{H}}_1 \hat{\mathbf{W}} \mathbf{u}_k \\ \mathbf{u}_{min} - \hat{\mathbf{H}}_2 (\hat{\xi}_1 - \tilde{\mathbf{y}}_{d,k+1}) - 2\hat{\mathbf{H}}_1 \hat{\mathbf{W}} \mathbf{u}_k \end{cases} \quad (40)$$

$$\hat{\mathbf{T}} = \begin{bmatrix} -\hat{\mathbf{H}}_1 & \hat{\mathbf{H}}_1 \\ \hat{\mathbf{H}}_1 & -\hat{\mathbf{H}}_1 \end{bmatrix}, \hat{\boldsymbol{\alpha}} = \begin{cases} \bar{\boldsymbol{\kappa}}_{k+1} \\ \underline{\boldsymbol{\kappa}}_{k+1} \end{cases}, \hat{\boldsymbol{\beta}} = \begin{cases} \bar{\boldsymbol{\sigma}}_{k+1} \\ \underline{\boldsymbol{\sigma}}_{k+1} \end{cases}$$

The LCP can be solved by many methods such as the interior point method [50], the non-interior continuation method [51], and the pivotal method [52]. In this article, the Lemke's algorithm that belongs to the category of the pivotal methods is used to solve the standard LCP (40). Thus, $\bar{\boldsymbol{\kappa}}_{k+1}$ and $\underline{\boldsymbol{\kappa}}_{k+1}$ can first be calculated by solving this LCP; then, substituting them into (36) and (28), the control inputs \mathbf{u}_{k+1} and the variable $\mathbf{z}_{k+1}^{(j)}$ can be obtained successively. So far, one iteration of the IOC controller has been completed. Thus, the current variables \mathbf{q}_{k+1} , $\boldsymbol{\lambda}_{k+1}$, and \mathbf{u}_{k+1} can be achieved in a few iterations. The convergence criterion can be defined based on the iterated variable \mathbf{z}_{k+1} or the function \mathcal{W} , given as

$$\|\mathbf{z}_{k+1}^{(j+1)} - \mathbf{z}_{k+1}^{(j)}\| / \|\mathbf{z}_{k+1}^{(j+1)}\| < \varepsilon, \text{ or } \|\mathcal{W}(\mathbf{z}_{k+1}^{(j+1)})\| < \varepsilon \quad (41)$$

where ε is the convergence error, and the former convergence criterion in (41) will be adopted in this work.

Finally, the velocities $\dot{\mathbf{q}}_{k+1}$ and the accelerations $\ddot{\mathbf{q}}_{k+1}$ can be

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

successively obtained by substituting \mathbf{q}_{k+1} and λ_{k+1} into (12)-(14). So far, all the unknown variables in the current time step $[t_k, t_{k+1}]$ have been obtained. Then, running step by step, the close-loop tracking control task of the TCR can be completed successfully. The pseudocode of the proposed DAEs model-based IOC framework for the position and orientation tracking control of the TCR has been summarized in Algorithm 1.

Remark 2: The control inputs of the proposed IOC algorithm are the driving speeds or lengths of cables rather than driving forces. This actuation mode will be more conducive to hardware implementation because, for most commercial motors, the accurate control forces or torques at the output end after the reducer are usually difficult to achieve.

Remark 3: The control input saturation, i.e., the inequality constraints, is transformed into an LCP. Therefore, the input saturation constraints can be satisfied directly just by solving the LCP, rather than by adjusting the control parameters, such as the weighting matrices, to indirectly guarantee that the actuation lengths or speeds are below their safety thresholds.

D. Stability Discussion of the Proposed IOC Method

System stability is always a crucial concern in evaluating the control performance of a controller. An unstable controller may supply inputs to the closed-loop plant to a point where the actuators fail or functional damage to the plant occurs. Thus, in this subsection, the stability of the proposed IOC method will be further investigated.

In the proposed IOC framework, the controlled dynamic system is discretized as (22), which implies a nonlinear discrete dynamical system. This closed-loop control system (22) with inputs \mathbf{u}_{k+1} is stable if the assumptions of the following theorem can be satisfied.

Theorem: Assume that all eigenvalues μ_i of the matrix Ψ_z lie inside the unit circle in the complex plane, i.e., $|\mu_i| < 1$ for all $i = 1, 2, \dots, n+m$. Then, locally, the close-loop control system governed by (22) is stable under the control law of the proposed IOC method. Here, Ψ_z is the Jacobian matrix of the state mapping function for the discrete system (22), given as

$$\Psi_z = \partial\mathbf{P}/\partial\mathbf{z} = \partial(\xi_1^{(j)} - \xi_2^{(j)} \cdot \mathbf{u}_{k+1})/\partial\mathbf{z}^{(j)} \quad (42)$$

Proof: The stability of a nonlinear discrete dynamic system can be analyzed by the fixed point stability theory [53]. First, substituting the suboptimal control inputs \mathbf{u}_{k+1} of (36) into (28), a map at time t_{k+1} for the nonlinear discrete system (22) can be established by a vector function $\Psi(\mathbf{z})$, which maps a vector $\mathbf{z}^{(j)}$ onto a vector $\mathbf{z}^{(j+1)}$, that is,

$$\mathbf{z}^{(j+1)} = \Psi(\mathbf{z}^{(j)}) \quad (43)$$

where

$$\begin{aligned} \Psi(\mathbf{z}^{(j)}) &= \xi_1^{(j)}(\mathbf{z}^{(j)}) + \xi_2^{(j)}(\mathbf{z}^{(j)}) \cdot [\hat{\mathbf{H}}_1(\mathbf{z}^{(j)}) \\ &(\bar{\mathbf{\kappa}}_{k+1} - \mathbf{\kappa}_{k+1} - 2\hat{\mathbf{W}}\mathbf{u}_k) - \hat{\mathbf{H}}_2(\mathbf{z}^{(j)}) \cdot (\hat{\xi}_1(\mathbf{z}^{(j)}) - \tilde{\mathbf{y}}_{d,k+1})] \end{aligned} \quad (44)$$

Then, the solution \mathbf{z}^* calculated by $\mathbf{z}^* = \Psi(\mathbf{z}^*)$ can be defined as a fixed point. Based on the stability theory of fixed points for discrete systems, the general result of the fixed point stability can be stated as that the convergence $\mathbf{z}^{(j)} \mapsto \mathbf{z}^*$ can only take place for $j \rightarrow \infty$ when $|\mu_i| < 1$. Herein, μ_i denotes all the eigenvalues of the Jacobian matrix Ψ_z , evaluated at the fixed point \mathbf{z}^* . Hence, if the fixed points of all the control steps are locally stable, the controlled trajectories including the position

and orientation will converge to a limited region around the desired trajectories in the whole time domain. That is to say, the proposed controller will be locally stable. Thus, the above theorem has been proven.

V. SIMULATION AND EXPERIMENTAL RESULTS

In this section, simulations and experiments on the position and orientation tracking control of the TCR were carried out to validate the proposed IOC approach. Firstly, the close-loop control platform was briefly introduced; and the effectiveness of the developed numerical model was evaluated. After that, the experimental results of position and orientation tracking control on the TCR were discussed in detail. Furthermore, the potential applications were also demonstrated by equipping the TCR with a laser or a visual device. The performance of our approach in the above experiments is also presented in the supplementary videos.

A. Experimental Setup for the Close-Loop Control

The closed-loop control experimental setup for the TCR is shown in Fig. 5. In the control loop, firstly, we got the feedback data, i.e., the current node positions \mathbf{q}_{k+1} of the TCR from a visual motion capture system named OptiTrack. For this visual system, the measurement error is less than 50 μm ; the average sampling period including node position capture and feedback communication is approximately 10 ms. The desired position and orientation data of the TCR's end were either written in a prepared file or generated from a user-defined function. Subsequently, a personal computer was used as the central control system to receive the above sensing and desired data, and to compute the suboptimal control inputs \mathbf{u}_{k+1} , i.e., the driving speeds for the next step of all sliding cables by the proposed IOC approach. Also, the driving lengths can be obtained, which are $\eta\mathbf{u}_{k+1}$. Finally, when the servo motors received the control signals through the Programmable Logic Controller (PLC) system, they would stretch or release the actuation lengths using transmission pulleys. After the above procedures, one control loop was finished, which was updated at 20 Hz in this work. Similarly, the entire closed-loop tracking control of the TCR can be completed step by step.

The data flow diagram of the closed-loop control is presented in Fig. 6. Herein, it should be noted that the current Lagrangian multiplier λ_{k+1} is not measured, but an approximate simulation result estimated by the IOC controller with taking the end's position and orientation from the feedback \mathbf{q}_{k+1} as the target. Specifically, when the current position \mathbf{q}_{k+1} is measured and sent to the controller, the real end's position and orientation from \mathbf{q}_{k+1} will be taken as the targets; and the proposed IOC algorithm will be executed once to update the simulation state of the TCR to approximate the real state. At the same time, λ_{k+1} can also be calculated. Then, both the measured \mathbf{q}_{k+1} and the estimated λ_{k+1} will be used as the feedback states \mathbf{z}_{k+1} and substituted into the equations of the LCP (39) and control law (36) to calculate the driving speeds \mathbf{u}_{k+1} for next step. In addition, it notes that all control computations were performed in MATLAB (R2021a) on a PC with an 11th Gen. Intel core i7-11700K CPU (3.60 GHz) processor and 16 GB RAM, running on Win10 64-bit.

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

TABLE II
POSITION COORDINATES OF THE TCR'S ENDPOINT FOR DIFFERENT CONFIGURATIONS

Coordinates	Configurations	Bending	Contraction	S-shape
Simulation \mathbf{P}_c (mm)		[80.66, -263.72, 464.26]	[62.59, -165.22, 473.28]	[47.24, 21.71, 501.87]
Experiment \mathbf{p}'_c (mm)		[73.75, -261.71, 466.62]	[63.14, -159.63, 471.19]	[49.88, 29.85, 506.55]
Absolute errors $\ \mathbf{p}_c - \mathbf{p}'_c\ $ (mm)		7.57	5.99	9.75
Relative errors defined by $\ \mathbf{p}_c - \mathbf{p}'_c\ / L$		1.26%	1.00%	1.63%
Relative errors defined by $\ \mathbf{p}_c - \mathbf{p}'_c\ / \ \mathbf{p}'_c - \mathbf{p}'_{c0}\ $		5.68%	6.58%	4.26%

Note: L here is the body length of the TCR, i.e., 0.6 m; $\mathbf{p}'_{c0} = [62.33, -160.89, 380.19]$ is the initial experimental end's position; and the driving lengths for the configurations of bending, contraction and S-shape are [90, 50, 50, 50, 50, 50] mm, [90, 70, 90, 70, 90, 70] mm and [60, 25, 55, 85, 55, 25] mm, respectively.

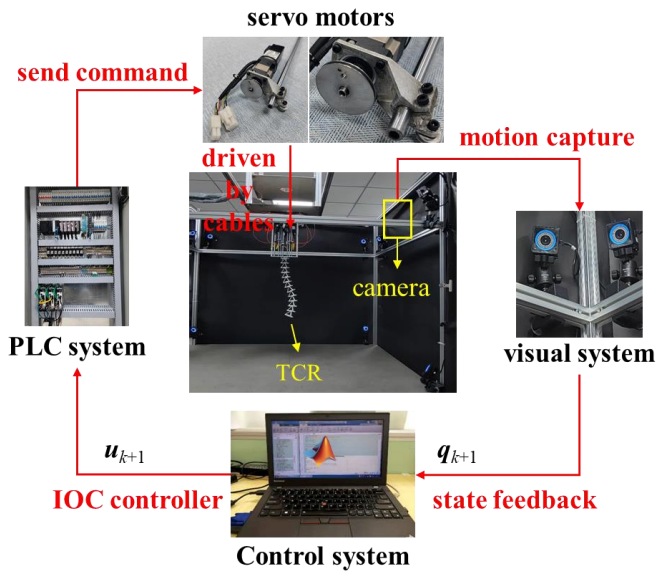


Fig. 5. Experimental setup for the closed-loop control of the TCR with the proposed IOC method.

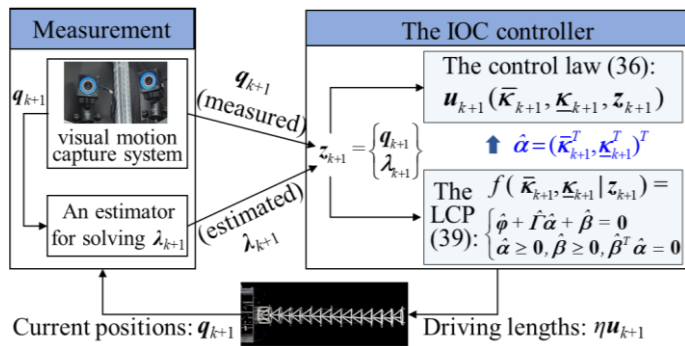


Fig. 6. Data flow diagram of the closed-loop control.

B. Model Evaluation

Using the model parameter identification algorithm in section III, the comprehensive errors in the experiment caused by the factors such as assembly, measurement, system friction, and production differences can intrinsically be compensated to some extent. Therefore, compared to the model established by theoretical parameters, the identified model can simulate the actual motion characteristics of the TCR more accurately. The geometric and material parameters of each component for the identified model are listed in Table I. Subsequently, to

evaluate the correctness of the identified model for the TCR system, certain groups of driving lengths that make the TCR contract, bend, or deform in an S-shape configuration are applied in a quasi-static way. The numerical results involving the end positions and body configurations are compared with those in the experiments.

As shown in Fig. 7, an S-shape configuration of the TCR under the driving lengths $l = [60, 25, 55, 85, 55, 25]$ mm and some snapshots in its deformation process are given, where the numerical results are all close to those by experiments. The average position errors of all the TCR's nodes at the driving lengths $(1/3)l$, $(2/3)l$ and l are 1.82 mm, 2.75 mm and 4.68 mm, respectively. Without loss of generality, the position errors of the TCR's endpoint for different configurations involving bending, contraction and S-shape deformation are further listed in Table II. Compared with the experimentally measured values, the absolute errors by numerical simulation are 7.57, 5.99, and 9.75 mm, respectively, which are 1.26%, 1% and 1.63% of the TCR's total length, or 5.68%, 6.58% and 4.26% of the corresponding end displacement.

In addition, to evaluate the dynamic error of the proposed DAEs model, a set of driving lengths $l = [10, 20, 150, 100, 10, 20]$ mm is applied on the TCR at a constant speed. The driving time is 10 s. As shown in Fig. 8, the configurations of the TCR at 0 s, 3 s, 7 s and 10 s, and the end position trajectories are all coincident between the results of simulations and experiments. The average position error is about 7.1 mm, which is 1.10% of the TCR's total length, or 1.32% of the end displacement.

Therefore, all the above comparison results indicate that our numerical model of the TCR for a controller design is available with enough accuracy.

C. Position Tracking Control of the Single-Segment TCR

Following a lot of research on trajectory tracking control of continuum robots, the end's position-only tracking control of the TCR is studied first in this subsection.

Using the proposed DAEs model-based IOC approach, five groups of experiments on the end's position tracking control of the TCR are conducted; the corresponding target trajectories are the curves like the letters "I ♥ DUT", which means "I Love Dalian University of Technology". Herein, the weighting matrices in (32) of the proposed IOC controller are set to $\hat{\mathbf{Q}} = 3.0 \times 10^4 \times \mathbf{I}_3$, $\hat{\mathbf{W}} = 3.5 \times 10^3 \times \mathbf{I}_3$, $\hat{\mathbf{R}} = 1.0 \times \mathbf{I}_3$, where

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

TABLE I
GEOMETRIC AND MATERIAL PARAMETERS OF THE TCR

Parameters	Value	Parameters	Value
Torus's diameter D_1 (mm)	70	Diameter of nylon rope D_2 (mm)	0.8
Cross-sectional diameter of rod A_r (mm)	5	Equivalent elastic modulus of nylon rope E_2 (MPa)	2300
Cell's average mass m_1 (g)	10	Spring's average mass m_2 (g)	0.8
Young's modulus of resin for 3D printing E_1 (MPa)	2500	Equivalent elastic modulus of spring E_3 (MPa)	90

Note: E_1 is theoretical value; E_2 and E_3 are identified values; and other parameters are measured values.

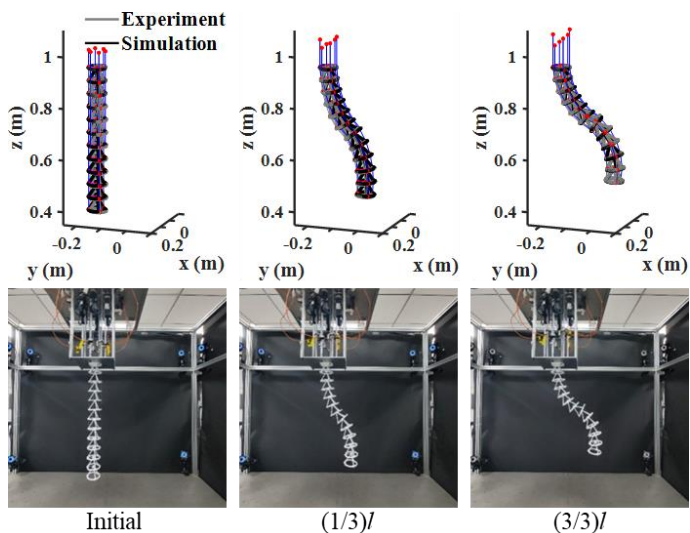


Fig. 7 Quasi-static model validation, where the driving lengths $l = [60, 25, 55, 85, 55, 25]$ mm

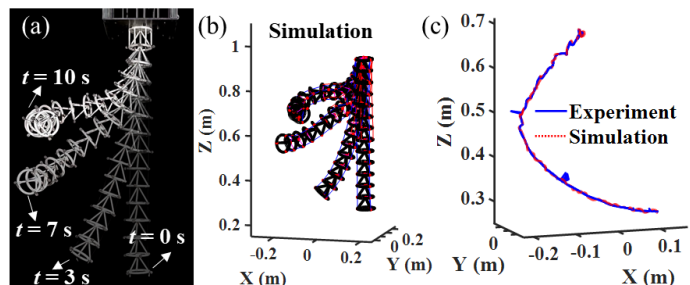


Fig. 8 Model validation of dynamics: (a-b) the configurations of experiment and simulation, (c) the end trajectories.

$I_3 \in \mathbb{R}^{3 \times 3}$ is the unit matrix. The convergence error ε is set to 2×10^{-6} . To ensure the physical safety of the servo motors, the input saturation constraints, i.e., the maximum driving speeds at each time step are limited between -50 mm/s and 50 mm/s (i.e., -50 mm/s $\leq \mathbf{u}_{k+1} \leq 50$ mm/s). Equivalently, the driving lengths at each time step are limited to $-2.5 \leq \eta \mathbf{u}_{k+1} \leq 2.5$ mm. However, the total driving length, calculated by algebraically summing the driving lengths from the initial step to the current control step, is not constrained here. In addition, since the spatial position of the TCR's endpoint can be controlled just by three active sliding cables, a single-segment TCR with

three sliding cable actuators is designed for the position tracking control task.

The control results with a T-shape target trajectory are presented in Fig. 9. In the experiment, to record and show the controlled endpoint trajectory more intuitively, a luminous ball is pasted at the bottom of the TCR's end. From Fig. 9(a), one can find that the luminous ball moves along the desired T-shape curve. The specific result data including the time history curves of the end position coordinates, the driving lengths and the tracking errors are given in Fig. 9(b-d).

As shown in Fig. 9(b), the position coordinate curves of the TCR's end by both simulation and experiment can follow the desired curves. But after partial amplification, it can still be seen that the control error of the experiment is larger. The maximum tracking error for simulation and experiment are approximately 2.30 mm and 8.57 mm, respectively, which are 0.38% and 1.43% of the TCR's total length. In addition, the driving lengths are further presented in Fig. 9(c).

To avoid repetitive statements, the experimental results of other position tracking control for drawing the letters "I ♥ DUT" are presented together in Fig. 10. As shown in Fig. 10(a1-a5), the actual trajectory curves of the TCR's end are always consistent with the desired curves. The time history curves of the corresponding tracking errors are given in Fig. 10(b1-b5). The maximum errors are 6.26 mm, 10.01 mm, 8.87 mm, 7.90 mm, and 8.57 mm, respectively.

Therefore, all the above numerical and experimental results illustrate that the proposed DAEs model-based IOC algorithm for the TCR can successfully complete the position tracking control tasks of "I ♥ DUT", which includes straight lines, right angles and arcs; and the average tracking error is less than 1 cm, which is approximately 1.67% of the TCR's body length.

D. Position and Orientation Collaborative Tracking Control of the Two-Segment TCR

The previous subsection has discussed the effectiveness of the proposed DAEs model-based IOC approach for the end's position tracking control of the TCR. Subsequently, several experiments on the position and orientation collaborative tracking control will be further conducted in this subsection using a two-segment TCR with 6 sliding cable actuators as presented in Fig. 2(a).

Without loss of generality, three control experiments on the position and orientation tracking control of the TCR have been conducted to further demonstrate the effectiveness of the proposed DAEs model-based IOC approach. Herein, the weighting matrices in (30) of the proposed IOC controller are set to $\hat{\mathbf{Q}} = 2.4 \times 10^4 \times I_9$, $\hat{\mathbf{W}} = 3.5 \times 10^3 \times I_6$, $\hat{\mathbf{R}} = 1.0 \times I_6$, where $I_9 \in \mathbb{R}^{9 \times 9}$ and $I_3 \in \mathbb{R}^{3 \times 3}$ are the unit matrices. In addition, it notes that two cells are added to the single-segment TCR by modular assembly so that the body length of this two-segment TCR is extended to 700 mm. For the convenience of expression here, these three experiments are named 'Exp_1', 'Exp_2' and 'Exp_3'. For the Exp_1, the TCR needs to do a fixed-point rotary motion at 55 degree angle to the horizontal plane. For the Exp_2 and the Exp_3, the end of the TCR needs to track a curve of cone frustum and a cylindrical helix, respectively with the required attitude.

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

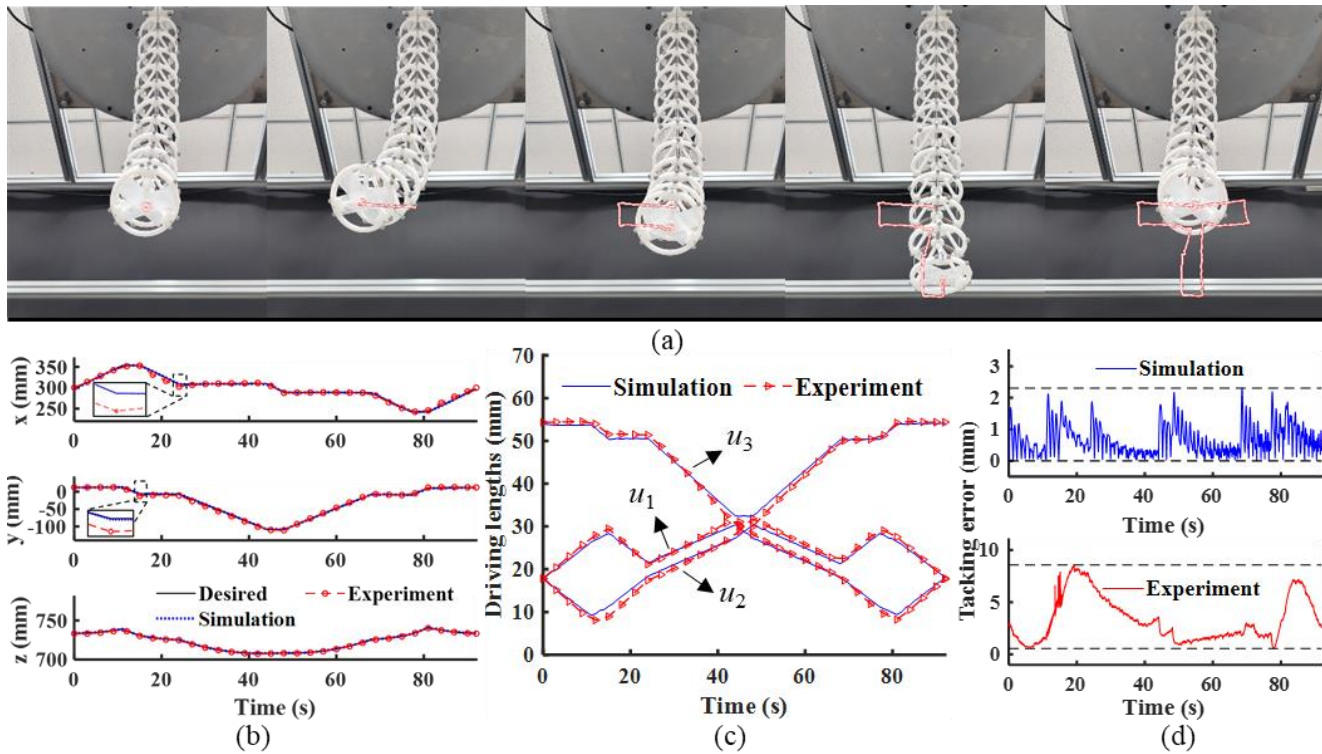


Fig. 9 The end's position tracking control results with T-shape target trajectory: (a) the deformation snapshots, (b-d) the time history curves of the end position coordinates, the driving lengths, and the tracking errors for simulations and experiments.

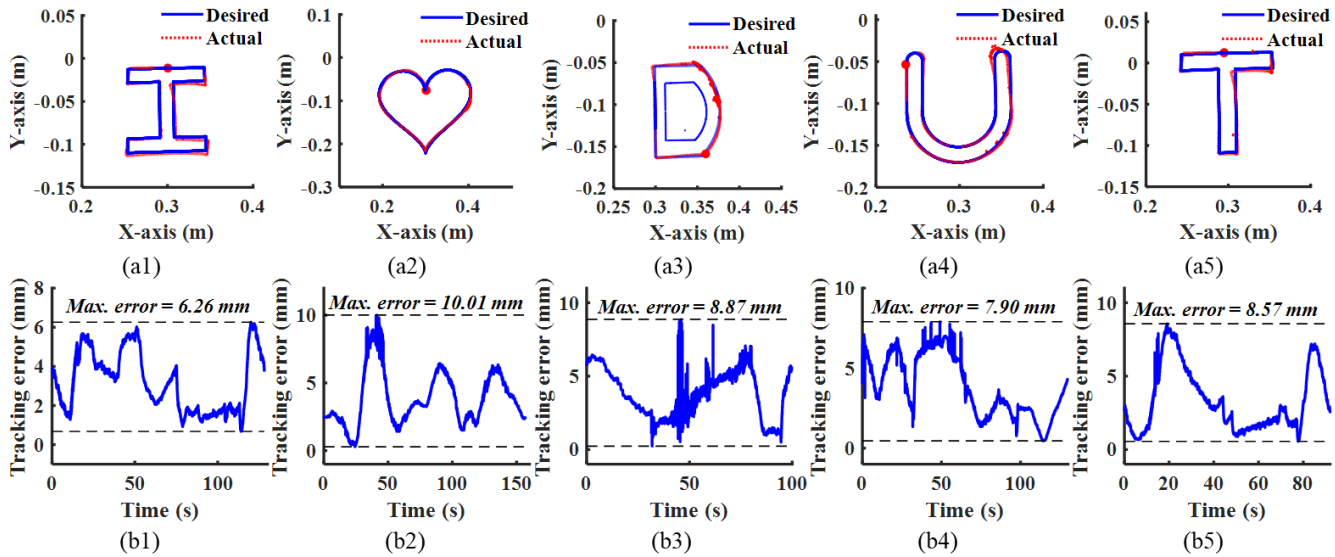


Fig. 10 The end's position tracking control results of the target trajectories "I ♥ DUT" by experiments: (a1-a5) the controlled trajectories of the TCR's endpoint position; (b) the time history curves of tracking errors.

As shown in Figs. 11(b1-b3), the desired and actual position trajectories of the TCR's end for the Exp_1, the Exp_2 and the Exp-3 are generally overlapping; the actual orientation vectors are given along the desired trajectories; and the driving lengths of the six active sliding cables are presented in Figs. 11(c1-c3). Furthermore, to show the tracking process intuitively, some configuration snapshots in the experiments are also provided in Figs. 11(a1-a3). Then, to quantify the orientation error more intuitively, the orientation of the end's normal vector can be equivalently represented by two rotation angles (θ_y and θ_x) around the global coordinate axes Y and X in sequence. Specifically, θ_x and θ_y can be solved by the nonlinear equation

$$\begin{cases} \sin(\theta_y) \\ -\cos(\theta_y) \cdot \sin(\theta_x) \\ \cos(\theta_y) \cdot \cos(\theta_x) \end{cases} - (r_3 - r_2) \otimes (r_2 - r_1) = \mathbf{0} \quad (45)$$

where r_1 , r_2 and r_3 are the positions of the end's three non-collinear nodes. Here, as shown in Fig. 12, the time history curves of the end orientation angles θ_x and θ_y are consistent with the desired curves. The maximum position errors of the Exp_1, the Exp_2 and the Exp-3 are 12.04 mm, 10.17 mm and 8.78 mm, respectively, which are about 1.7%, 1.5% and 1.3% of the TCR's body length. The maximum absolute errors of

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

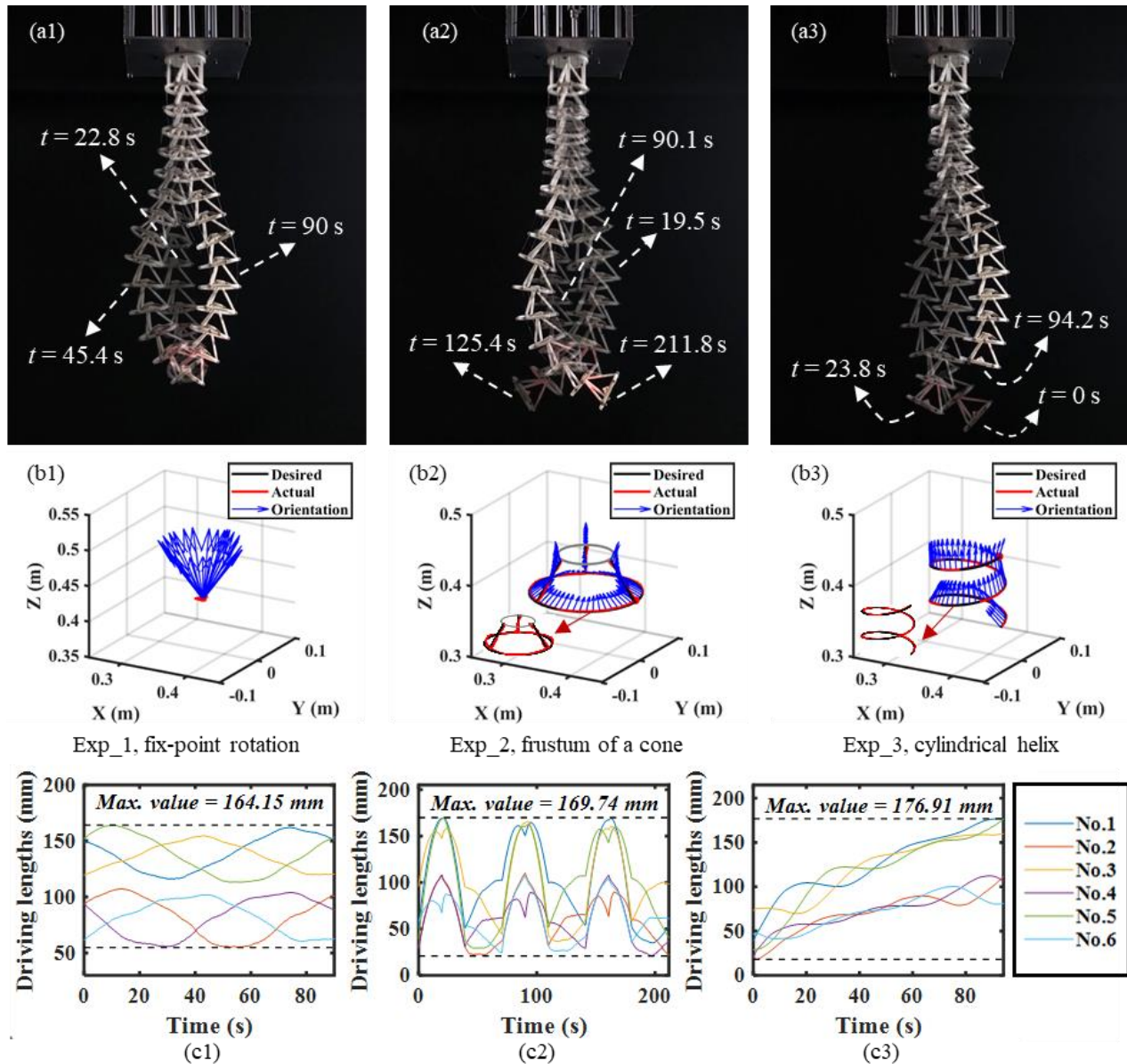


Fig. 11 Position and orientation tracking control of the TCR's end for the experiments with different special curves (i.e., named Exp-1, Exp-2, and Exp-3, respectively): (a1-a3) the snapshots, (b1-b3) the controlled trajectories (i.e., a fixed point, a curve of cone frustum, and a cylindrical helix) and orientation (the blue arrows), (c1-c3) the driving lengths.

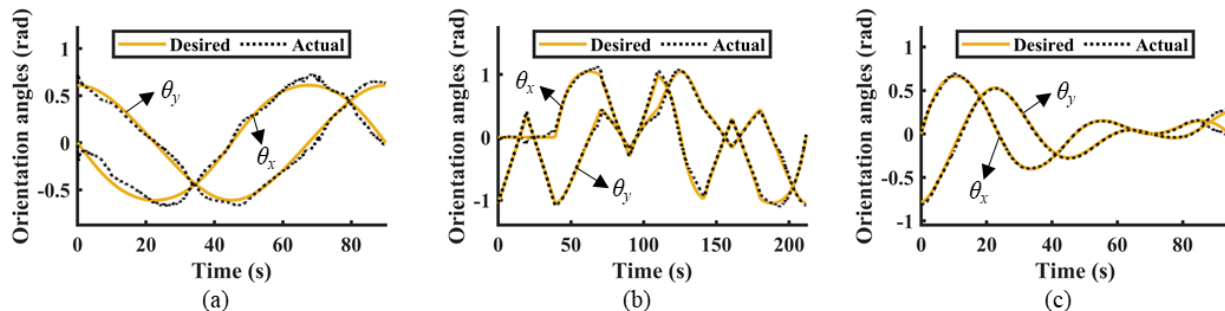


Fig. 12 The end's orientation angles of the position and orientation tracking controls for (a) Exp_1, (b) Exp_2 and (c) Exp_3.

the orientation angles θ_x and θ_y are about 0.181 rad, 0.141 rad, and 0.087 rad, respectively.

Furthermore, to illustrate the control stability discussed in sub-section IV (E), numerical simulations are carried out to analyze the stability of all the fixed points for the nonlinear

discrete map $\Psi(z)$ in different tracking task (Exp- k , where $k = 1, 2, 3$) at each time step. Figure 13 shows the maximum absolute (Max. Abs.) eigenvalues of the Jacobian matrix Ψ_z , evaluated at each solution point, i.e., the fixed point. From Fig. 13, it can be deduced that all the eigenvalues at each step will

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

TABLE III
CONTROL PERFORMANCE COMPARISON BETWEEN OUR STUDY AND THE EXISTING WORKS

Performance Reference	Robot structures (Initial length L)	Control accuracy			Computation time (software)	Model-based or model-free	Open or closed loop
		Position error ϵ_p	Relative position error	Orientation angle error			
Ref. [22]	Magnetic-driven soft Microcatheters ($L = 70$ mm)	1.3 ± 0.48 mm	$< 2.5\%$	$2.8 \pm 1.0^\circ$	6.5 ms per inverse step (Matlab)	Inverse kinematics model-based	Closed loop
Ref. [23]	Pneumatic-driven hyperelastic continuum robots ($L = 90$ mm)	< 4.5 mm	$< 5\%$	RMSE: $2.0^\circ \sim 3.1^\circ$	77 ms per inverse step (Matlab)	Inverse statics (Cosserat rod model) model- based	Open loop
Ref. [24]	Pneumatic-driven 3- DOF soft manipulator ($L = 536$ mm) Cable-driven	< 13.4 mm	$< 2.5\%$	\	8.2 ms per inverse step (Matlab)	Inverse kinematics model-based	Closed loop
Our work	tensegrity continuum robot (TCR) ($L = 700$ mm)	< 12.0 mm	$< 1.7\%$	RMSE: $1.1^\circ \sim 3.0^\circ$	Less than 10 ms per sampling step (Matlab)	Dynamics (DAEs) model-based	Closed loop

The relative position error is equal to ϵ_p/L ; RMSE refers to the root mean square error; ‘\’ denotes that the orientation control is not considered.

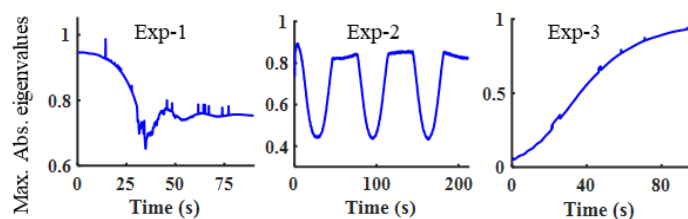


Fig. 13 Max. Abs. eigenvalues for different tracking tasks of Exp-1, Exp-2, and Exp-3.

lie in a unit circle, since their Max. Abs. values are less than one throughout the entire time domain. Based on the stability theory of fixed points [53], the states controlled by the proposed method are stable and reliable.

In addition, some key performance indicators of end's position and orientation control for continuum robots found in the existing works [22]-[24] and our study has been summarized in Table III. As shown in Table III, the robot structures (material and initial length) and actuation manners among all works are different, but the initial length of the TCR is the longest. For the control accuracy, despite the maximum position error in our work is slightly larger than that of [22] and [23], the relative position error of our work is the smallest, and the root mean square error (RMSE) of the orientation angle error is also smaller than those in the existing literature. For the computational speed, our work can calculate at a rate higher than the average compared to the other methods. Thus, based on these comparison results, the proposed approach has better comprehensive control performances.

Overall, all the above experimental results and discussions illustrate that the proposed DAEs model-based IOC approach can achieve high-precision tracking control of both the end's position and orientation for complex planar or spatial target trajectories involving fixed points, straight lines, arcs, bending angles, and right angles.

E. Virtual experiments on fast dynamic tracking control

The closed-loop controls in Sub-sections V(C-D) were implemented at a lower frequency of 20 Hz. In fact, the online computational time of the proposed IOC approach is only the

time for solving a small-scale (12-dimensional) LCP problem, which is highly efficient. The calculation time of the controller for each sampling period is under 10 ms, enabling a potential control frequency of up to 100 Hz. However, due to the constraints posed by the current experimental hardware, such as much time costs associated with measurement, motor drive, and data communication, the control frequency has to be limited, and the TCR should move at relatively low speeds. To further demonstrate the effectiveness of the proposed method for fast dynamic tracking control, additional virtual closed-loop control experiments are conducted in this subsection.

In the virtual experiments, the closed-loop interaction between the virtual sensor and the IOC controller is presented in Fig. 14. The validated dynamic DAEs model (Model A) is used as a virtual sensor for state feedback. The close-loop controller is established based on another model (Model B), which can be a kinematic, static, or dynamic system. To account for the inevitable motion errors of the TCR between virtual simulation and physical prototype, certain material parameters in Model B are set differently from those in Model A. Specifically, in Model B, the elastic modulus of the classical cables is reduced by 30%, and the density of all components is increased by 20%.

Then, as shown in Fig. 15, a fixed-point orientation tracking control problem is taken as an example to illustrate the effectiveness of the proposed method for fast dynamic tracking. Here, a comparison between a static model-based controller and a dynamic model-based controller at different motion rates are provided in Figs. 16(a-c), where the upper half are the time history curves of the end's position in X-direction from fast to slow rate, and the lower half are that of the end's orientation angle θ_y .

For the static model-based controller, it notes that the static model here refers to a model where the first dynamic equation in the DAEs of (9) is degraded to a static form. As shown in Figs. 16(a-b), when the TCR moves at a fast speed (the total time of 10 s) or at a medium speed (total time of 45 s), both the end's position and orientation angle will quickly deviate from the desired trajectories, ultimately leading to divergence.

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

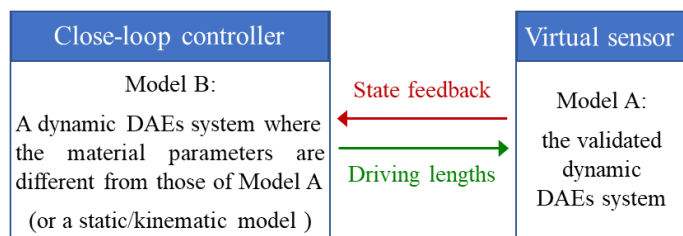


Fig. 14. The framework of the virtual control experiment.

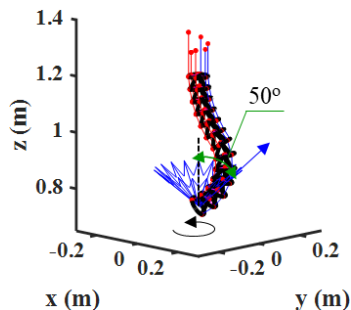


Fig. 15 The fixed-point orientation tracking control problem at different motion rates.

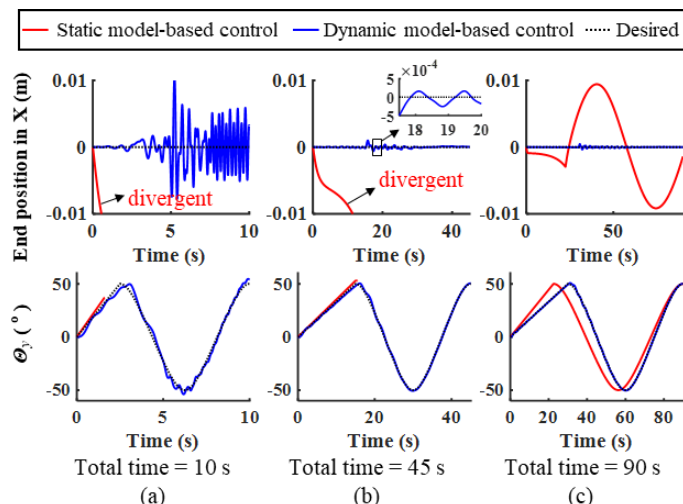


Fig. 16. Control result comparisons between the static model-based controller and the dynamic model-based controller at different tracking rates: (a-c) the tracking times are 10 s (fast), 45 s (medium) and 90 s (slow), respectively.

As shown in Fig. 16(c), the tracking task can only be completed when the total motion time is extended to 90 s. However, even then, the control error remains significant, with maximum absolute errors of 9.43 mm for the end position in X-direction and 25.29° for the end orientation angle θ_y .

For the dynamic model-based controller, as shown in Fig. 16, it can effectively track the desired end's position and orientation at different rates, but the control accuracy will decrease as the tracking rate increases due to the heightened dynamic effect of the TCR. Specifically, the root mean square errors (RMSE) of the end position in X-direction for the tracking times 10 s, 45 s and 90 s are 2.56 mm, 0.16 mm, and 0.07 mm, respectively; and their maximum absolute errors are 10.15 mm, 1.0 mm and 0.47 mm, respectively. The RMSE of the end orientation angle θ_y are 2.98° , 1.14° and 0.49° , respectively; and their maximum absolute errors are 5.84° , 4.32° , 2.158° and 2.45° , respectively.

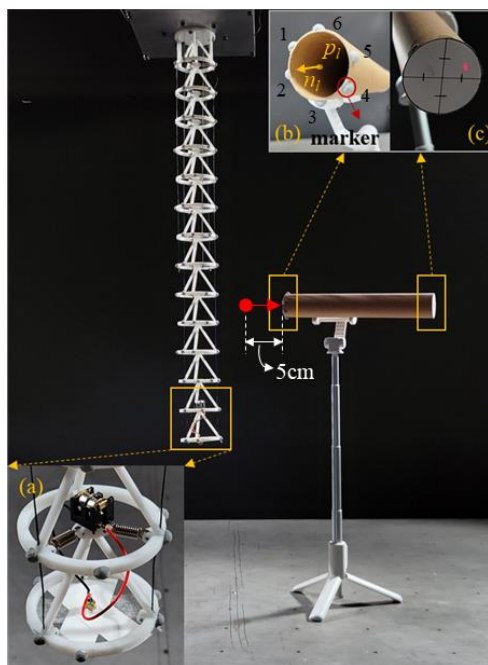


Fig. 17 Scenario 1, a laser passes through a slender pipeline: (a) the laser device, (b) position and attitude determination of the slender pipeline, (c) the Laser spot.

All the above results illustrate that the proposed DAEs-model based IOC approach is effective not only for the end's position and orientation tracking control of the TCR in slow-motion scenarios but also for fast dynamic tracking control.

F. Potential Applications in Industrial Inspection

When equipped with certain devices, the TCR can be used as an autonomous auxiliary tool for industrial in-situ detection in narrow environments due to its dexterity in achieving both orientation and position control. To demonstrate this potential application, a laser or a camera was installed at the end, and two scenarios were designed. The experimental results are discussed as follows.

Scenario 1: as shown in Fig. 17, a laser was installed at the TCR's end; the control task is to make the laser beam pass through a slender pipeline. First, the position of the pipeline's left endpoint P_l , i.e., the expected laser incident point, and the orientation of the pipeline's central axis n_l , i.e., the expected laser incidence direction can both be calculated using the positions of the six symmetrically distributed markers on the pipeline shown in Fig. 17(b), which were captured by the visual motion capture system OptiTrack. Then, the direction vector n_l and the point that 5 cm away from P_l along the pipeline axis were chosen as the target point and the target orientation to be tracked, respectively. During the experiments, both the trajectory and orientation between the start point and the target point were discretized with a proper step length. Subsequently, the proposed IOC method was used to solve the tracking control problem based on the above defined trajectory and orientation. Finally, the laser would project through the pipeline onto the pre-pasted surface at the right end shown in Fig. 17(c). The control results are presented in Figs. 18 and 19.

Figure 18 shows some snapshots in the tracking process. When the TCR's end reaches the first target point P_l , the laser

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

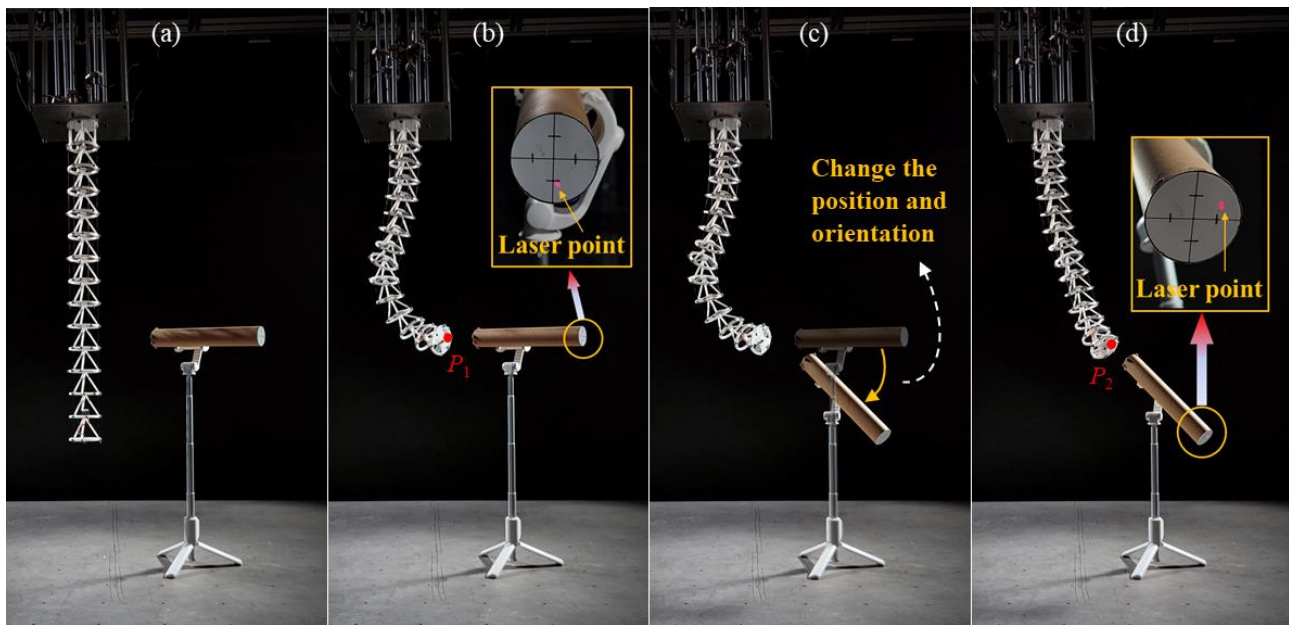


Fig. 18 Experimental snapshots in the tracking process: (a) the initial state, (b) configuration and laser point at the first target P_1 , (c) moving the pipeline to get a new target position and orientation, (d) configuration and laser point at the second target P_2 .

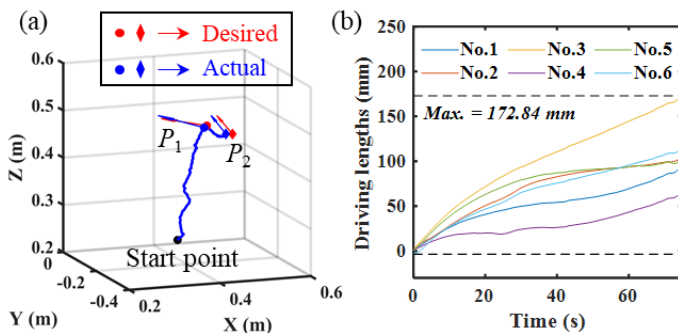


Fig. 19 The control results of scenario 1: (a) the position and orientation, (b) the driving lengths.

will pass through the pipeline placed horizontally, and project onto the surface at the right end. Afterward, the attitude of the pipeline is changed to be inclined. And the TCR continues to perform tracking control tasks with this new target position and orientation. When it reaches the second target point P_2 , the laser can also pass through the inclined pipeline. The desired and actual position and orientation at the target points are presented in Fig. 19(a). The position errors at the target point P_1 and P_2 are 8.7 mm and 13.9 mm, respectively, which are about 1.24% and 1.99% of the two-segment TCR's body length. And the absolute errors of the orientation angles θ_x and θ_y are about 0.161 rad and 0.038 rad at point P_1 , and 0.024 rad and 0.033 rad at point P_2 . In addition, the driving lengths are presented in Fig. 19(b).

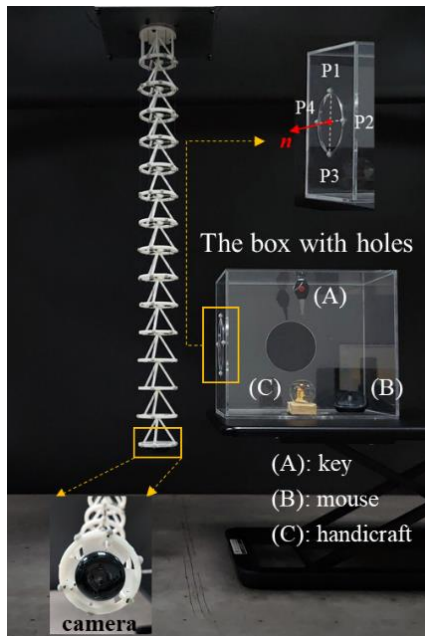


Fig. 20 Scenario 2, internal environment detection of a closed box with one or more narrow inlets.

Scenario 2: as shown in Fig. 20, by replacing the laser in scenario 1, a camera was installed at the TCR's end; the control task is to detect the internal environment of a closed box with one or more narrow inlets to find three given target objects. First, the position and attitude of the closed box should be calculated using the positions of the symmetrically distributed markers P_i ($i = 1, 2, 3, 4$) around the left hole of the box. Then, a target trajectory needs to be created. Finally, one can enable the end camera and start the tracking control task. Herein, it should be clarified that the goal of this scenario is not to recognize objects, but to show how orientation control affects detection performance.

For the above control task, to demonstrate the merits of the proposed controller with the TCR's end orientation control, two detection strategies with and without orientation control will be compared in this scenario. As presented in Fig. 21(a), the main idea of Strategy 1 is to directly try multiple times through different inlet holes without orientation control; the trajectory is designed as a polyline. However, as shown in Fig. 21(b), Strategy 2 tries to complete the detection task by scanning the internal environment through only one nearby inlet hole with orientation control.

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <



Fig. 22 Experimental snapshots for Strategy 1 from the main view, the top view and the camera at the TCR's end.

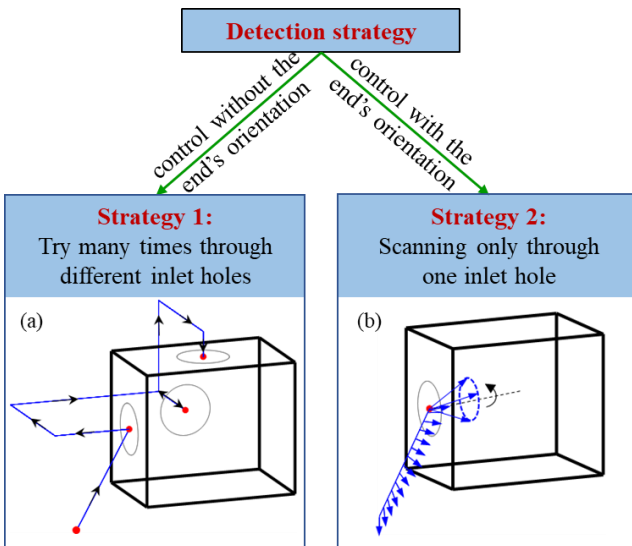


Fig. 21 The detection strategies without (Strategy 1) or with (Strategy 2) end's orientation control in Scenario 2.

Here, it is noted that the box only serves as a confined space; its transparency does not affect the validation of the above proposal since the TCR does not enter the interior of the box. When the TCR needs to enter the box for inspection, it may be impractical to use a visual motion capture system that is not integrated within the robot for state feedback. For this situation, a sensor such as fiber Bragg grating (FBG) could potentially be integrated onto the TCR for state feedback.

Before discussing the control results, it is worth noting that for Strategy 1, despite there is no need to consider the end's orientation in its controller design, obviously, multiple inlet holes of different directions are required. This requirement may not be met for some industrial equipment to be detected.

Even worse, from some experiments by Strategy 1, we found that the end's attitude at the same target point was related to its initial attitudes and the tracked paths. This correlation will make the controller heavily dependent on manual prediction. Then, the control results using the above two strategies have been compared as follows.

For Strategy 1, Fig. 22 presented some snapshots from the main view, the top view and the camera at the end synchronously. From Fig. 22, it can be seen that the TCR can successively reach the first two inlet holes on the left and the back of the closed box along the desired polyline. When the TCR reached the 2nd hole at time 133.1 s, a target object, i.e., the mouse was detected. However, before reaching the 3rd inlet hole, the TCR collided with the edge of the closed box at time 180 s, so the tracking control task was terminated here. In addition, the end's position trajectory and the time history curves of the driving lengths were given in Fig. 23.

For Strategy 2, the control process can be divided into four stages. In the 1st stage, as shown in Fig. 24(a), the TCR's end reached the inlet hole along a straight line. Synchronously, the end's orientation gradually changed from the initial state to perpendicular to the inlet hole. Then, the TCR began to scan the internal environment of the box at the target point. In the 2nd stage, the TCR continued to adjust its end's orientation to an angle of $\pi/6$ with the X-axis and rotated once around it in the 3rd stage. Afterward, returned to perpendicular to the inlet hole in the 4th stage. Figs. 24(b-d) show the snapshots of the moments when the target objects were detected. The camera's orientations of the TCR for these snapshots were also given in Fig. 24(e). The tracking control results including the end's position and orientation were shown in Fig. 25. From Fig. 25, it can be seen that the actual end's position in Z-direction and the normal vector of the four stages were both consistent with

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

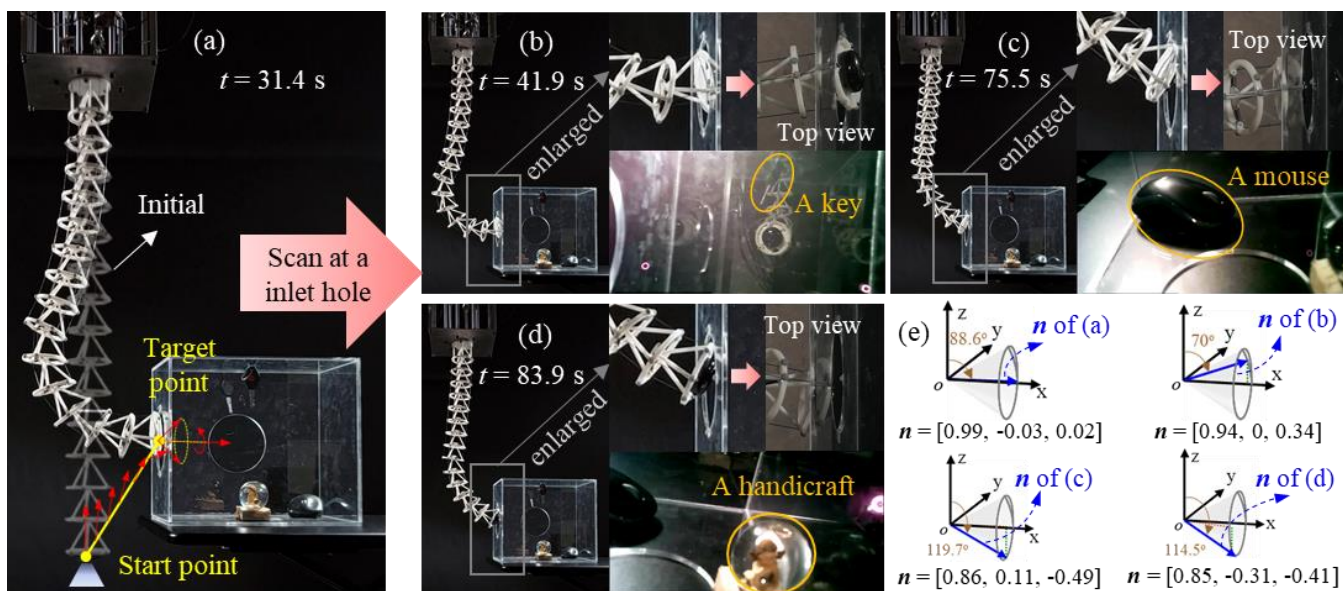


Fig. 24 Experimental snapshots for Strategy 2: (a) the snapshot at the time of starting the scan, (b-d) the snapshots at the time of detecting the target objects, (e) the camera's orientations of the TCR in these snapshots, which were opposite to that of the end.

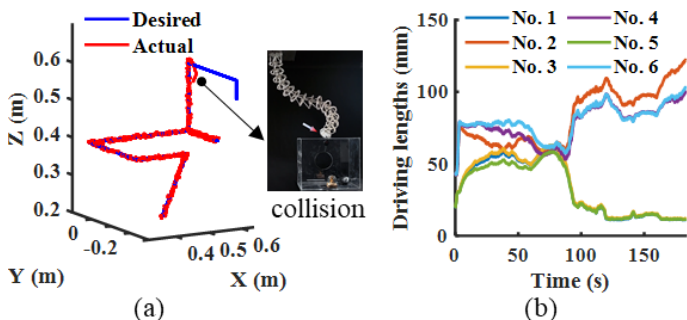


Fig. 23 The control results of Strategy 1: (a) the tracking trajectories, (b) the driving lengths.

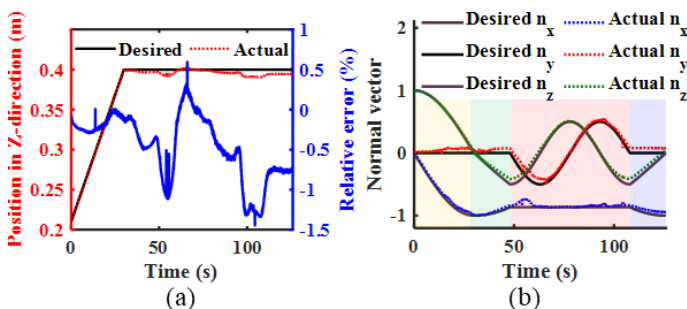





Fig. 25 The control results of Strategy 2: (a) the position coordinate in the Z-direction and its relative errors, (b) the end's normal vector coordinates.

TABLE IV
LIST OF THE DETECTED OBJECTS

Target Objects			
Has it been detected? (Yes or No)	Key	Mouse	Handcraft
Strategy 1	No	Yes	No
Strategy 2	Yes	Yes	Yes

the desired ones. In addition, the detected objects by the above two strategies are listed in Table IV. When using the position and orientation collaborative tracking control strategy, i.e., Strategy 2, all the three target objects can be detected, and the detection time was about 84 s. But only one target object was detected by Strategy 1 without orientation control; and the detection time was even longer, which was 183 s for this task.

In sum, all the above experimental results and discussions illustrate that the proposed DAEs model-based IOC approach is effective, practical and easy to implement for the position and orientation tracking control of the cable-driven tensegrity continuum robots. It enhances the control performance of the traditional position-only tracking controllers for the continuum robots and meets the demands of more application scenarios. In addition, the proposed controller is more compatible with the working modes of most current commercial motors by combining a so-called geometric constraint driven technology, where the speeds or lengths of the sliding cables serve as the control inputs rather than their forces or torques.

VI. CONCLUSION

In this article, a DAEs model-based IOC approach was proposed to solve the position and orientation tracking control problem of a tensegrity continuum robot with sliding cable actuators. First, combining the geometrically nonlinear PFEM and the multibody dynamics theory, a control-oriented model of the TCR described by DAEs was established, where the absolute nodes' positions and the sliding cables' lengths (or speeds) serve as the basic state variables and the control inputs, respectively. So that the feedback of the current configuration of the TCR can be measured directly by visual equipment, without any data conversion or analysis; and the control inputs are compatible with the operating modes of most commercial motors. Afterward, the position and orientation cooperative tracking control problem was described as a nonlinear optimal control problem with DAEs and input saturation constraints. Then, to overcome the

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

difficulties of solving this problem, it was transformed into a sequence of suboptimal IOC problems at every discrete time slot. Finally, the control inputs can be achieved by solving these IOC problems. The numerical and experimental findings demonstrated that the proposed DAEs model-based IOC method is effective, and can significantly improve the control performance of the position-only tracking controllers and meet the requirements of more application scenarios. In addition, it is worth mentioning that the proposed IOC tracking control framework is not only applicable to the TCR in this work, but can also be extended to other cable-driven continuum robot systems described by the DAEs model.

REFERENCES

- [1] G. Robinson and J. B. C. Davies, "Continuum robots - a state of the art," in *Proc. IEEE Int. Conf. Robot. Autom.*, Detroit, Michigan, USA, May 1999, pp. 2849-2854.
- [2] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *J. Robot. Syst.*, vol. 20, no. 2, pp. 45-63, Feb. 2003.
- [3] M. Russo, S. M. H. Sadati, X. Dong, A. Mohammad, I. D. Walker, C. Bergeles, K. Xu, and D. A. Axinte, "Continuum robots: An overview," *Adv. Intell. Syst.*, vol. 5, no. 5, 2023.
- [4] Z. Z. Liu, Z. Q. Cai, H. J. Peng, X. G. Zhang, and Z. G. Wu, "Morphology and tension perception of cable-driven continuum robots," *IEEE-ASME Trans. Mechatron.*, vol. 28, no. 1, pp. 314-325, Feb. 2023.
- [5] M. Wang, X. Dong, W. Ba, A. Mohammad, D. Axinte, and A. Norton, "Design, modelling and validation of a novel extra slender continuum robot for in-situ inspection and repair in aeroengine," *Robot. Comput. Integr. Manuf.*, vol. 67, pp. 102054, 2021.
- [6] G. P. Scott, C. G. Henshaw, I. D. Walker, and B. Willimon, "Autonomous robotic refueling of an unmanned surface vehicle in varying sea states," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Hamburg, Germany, 2015, pp. 1664-1671.
- [7] A. Mohammad, M. Russo, Y. H. Fang, X. Dong, D. Axinte, and J. Kell, "An efficient follow-the-leader strategy for continuum robot navigation and coiling," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7493-7500, Oct. 2021.
- [8] C. D. Santina, C. Duriez, and D. Rus, "Model based control of soft robots: a survey of the state of the art and open challenges," *IEEE Control Syst. Mag.*, vol. 43, no. 3, pp. 30-65, Jun. 2023.
- [9] X. M. Wang, Y. Q. Li, and K. W. Kwok, "A survey for machine learning-based control of continuum robots," *Front. Robot. AI*, vol. 8, pp. 730330, 2021.
- [10] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, and C. Laschi, "Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature," *IEEE Trans. Robot.*, vol. 31, no. 4, pp. 823-834, Aug. 2015.
- [11] R. J. Webster, and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1661-1683, Nov. 2010.
- [12] Z. G. Mu, Y. Q. Chen, Z. Li, C. J. Wang, N. Ding, and H. H. Qian, "A combined planning method based on Biaric curve and Bézier curve for concentric cable-driven manipulators working in confined environments," *IEEE-ASME Trans. Mechatron.*, vol. 27, no. 6, pp. 4475-4486, Dec. 2022.
- [13] T. Greigarn, and M. C. Çavusoglu, "Pseudo-rigid-body model and kinematic analysis of MRI-actuated catheters," 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 2236-2243, 2015.
- [14] I. A. Gravagne, and I. D. Walker, "Uniform regulation of a multi-section continuum manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, USA, May 2002, pp. 1519-1524.
- [15] C. Della Santina, R. K. Katschmann, A. Bicchi, and D. Rus, "Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment," *Int. J. Robot. Res.*, vol. 39, no. 4, pp. 490-513, Mar. 2020.
- [16] A. Kazempour, O. Fischer, Y. Toshimitsu, K. W. Wong, and R. K. Katschmann, "Adaptive dynamic sliding mode control of soft continuum manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, Philadelphia, PA, USA, May. 2022, pp. 3259-3265.
- [17] A. A. Alqumsan, S. Khoo, and M. Norton, "Robust control of continuum robots using Cosserat rod theory," *Mech. Mach. Theory.*, vol. 131, pp. 48-61, 2019.
- [18] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, "Koopman-based control of a soft continuum manipulator under variable loading conditions," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6852-6859, 2021.
- [19] A. P. Sabelhaus, H. Zhao, E. L. Zhu, A. K. Agogino, and A. M. Agogino, "Model-predictive control with inverse statics optimization for tensegrity spine robots," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 1, pp. 263-277, Jan. 2021.
- [20] H. J. Peng, H. Yang, F. Li, C. Z. Yang, and N. N. Song, "A unified framework for mechanical modeling and control of tensegrity robots," *Mech. Mach. Theory.*, vol. 191, pp. 105498, 2024.
- [21] L. Tang, J. Huang, L.-M. Zhu, X. Zhu, and G. Gu, "Path tracking of a cable-driven snake robot with a two-level motion planning method," *IEEE-ASME Trans. Mechatron.*, vol. 24, no. 3, pp. 935-946, 2019.
- [22] D. J. Lin, W. Y. Chen, K. He, N. D. Jiao, Z. D. Wang, and L. Q. Liu, "Position and orientation control of multisection magnetic soft microcatheters," *IEEE-Asme Transactions on Mechatronics*, vol. 28, no. 2, pp. 907-918, Apr. 2023.
- [23] J. L. Shi, S. A. Abad, J. S. Dai, and H. A. Wurdemann, "Position and orientation control for hyperelastic multisegment continuum robots," *IEEE-ASME Trans. Mechatron.*, vol. 29, no. 2, pp. 995-1006, Apr. 2024.
- [24] Z. Y. Gong, X. Fang, X. Y. Chen, J. H. Cheng, Z. X. Xie, J. Q. Liu, B. H. Chen, H. Yang, S. H. Kong, Y. F. Hao, T. M. Wang, J. Z. Yu, and L. Wen, "A soft manipulator for efficient delicate grasping in shallow water: Modeling, control, and real-world experiments," *International Journal of Robotics Research*, vol. 40, no. 1, pp. 449-469, Jan. 2021.
- [25] Z. Y. Dong *et al.*, "Shape tracking and feedback control of cardiac catheter using MRI-guided robotic platform-validation with pulmonary vein isolation simulator in MRI," *IEEE Trans. Robot.*, vol. 38, no. 5, pp. 2781-2798, Oct. 2022.
- [26] D. Trivedi, A. Lotfi, and C. D. Rahn, "Geometrically exact models for soft robotic manipulators," *IEEE Trans. Robot.*, vol. 24, no. 4, pp. 773-780, 2008.
- [27] R. E. Skelton, and M. C. de Oliverira, "Introduction and motivation" in *Tensegrity Systems*, New York, USA: Springer, 2009, pp. 1-43.
- [28] K. J. Bathe, "Formulation of the finite element method—Linear analysis in solid and structural mechanics," in *Finite Element Procedures*, New Jersey, USA: Prentice-Hall, 1996, pp. 148-214.
- [29] R. E. Skelton, Efficient models of multi-body dynamics. In: Blockley, R., Shyy, W., Encyclopedia of Aerospace Engineering. New York, USA: John Wiley & Sons, Ltd. 2010.
- [30] Z. Y. Kan, H. J. Peng, B. Chen, and W. Zhong, "Nonlinear dynamic and deployment analysis of clustered tensegrity structures using a positional formulation FEM," *Compos. Struct.*, vol. 187, pp. 241-258, Mar. 2018.
- [31] T. M. Bieze, A. Kruszewski, B. Carrez, and C. Duriez, "Design, implementation, and control of a deformable manipulator robot based on a compliant spine," *Int. J. Robot. Res.*, vol. 39, no. 14, pp. 1604-1619, Dec. 2020.
- [32] Z. Wang, G. Wang, X. Chen, and N. M. Freris, "Dynamical modeling and control of soft robots with non-constant curvature deformation," 2022, *arXiv: 2203.07929*.
- [33] M. Li, R. Kang, S. Geng, and E. Guglielmino, "Design and control of a tendon-driven continuum robot," *Trans. Inst. Meas. Control*, vol. 40, no. 11, pp. 3263-3272, 2017.
- [34] M. Thieffry, A. Kruszewski, T.-M. Guerra, and C. Duriez, "Trajectory tracking control design for large-scale linear dynamical systems with applications to soft robotics," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 2, pp. 556-566, 2021.
- [35] J. N. Yang, A. Akbarpour, and P. Ghaemmaghami, "New optimal control algorithms for structural control," *J. Eng. Mech.*, vol. 113, no. 9, pp. 1369-1386, 1987.
- [36] J. Zhang, et al., "A preprogrammable continuum robot inspired by elephant trunk for dexterous manipulation," *Soft Robot.*, vol. 10, no. 3, pp. 636-646, Jan. 2023.
- [37] Z. Littlefield, D. Surovik, M. Vespignani, J. Bruce, W. Wang, and K. E. Bekris, "Kinodynamic planning for spherical tensegrity locomotion with effective gait primitives," *Int. J. Robot. Res.*, vol. 38, no. 12-13, pp. 1442-1462, Oct. 2019.
- [38] H. Lee, Y. Jang, J. K. Choe, S. Lee, H. Song, J. P. Lee, N. Lone, and J. Kim, "3D-printed programmable tensegrity for soft robotics," *Sci. Robot.*, vol. 5, no. 45, Aug. 2020.

> REPLACE THIS LINE WITH YOUR MANUSCRIPT ID NUMBER (DOUBLE-CLICK HERE TO EDIT) <

- [39] S. Ikemoto, K. Tsukamoto, and Y. Yoshimitsu, "Development of a modular tensegrity robot arm capable of continuous bending," *Frontiers in robotics and AI*, vol. 8, pp. 774253-774253, 2021.
- [40] V. Ramadoss, K. Sagar, M. S. Ikkal, J. H. L. Calles, R. Siddaraboina, and M. Zoppi, "Hedra: A bio-inspired modular tensegrity robot with polyhedral parallel modules." *IEEE 5th International Conference on Soft Robotics (RoboSoft)* pp. 559-564, 2022.
- [41] D. Zappetti, R. Arandes, E. Ajanic, and D. Floreano, "Variable-stiffness tensegrity spine," *Smart Materials and Structures*, vol. 29, no. 7, Jul, 2020.
- [42] H. C. Tran, and J. Lee, "Geometric and material nonlinear analysis of tensegrity structures," *Acta Mechanica Sinica*, vol. 27, no. 6, pp. 938-949, 2011.
- [43] L. Zhang, M. K. Lu, H. W. Zhang, and B. Yan, "Geometrically nonlinear elasto-plastic analysis of clustered tensegrity based on the corotational approach," *Int. J. Mech. Sci.*, vol. 93, pp. 154-165, 2015.
- [44] J. T. Betts, "The optimal control problem" in *Practical Methods for Optimal Control Using Nonlinear Programming*, 2nd ed., Philadelphia, USA: SIAM, 2010, pp. 81-125.
- [45] M. Gerdt, "Necessary conditions for infinite optimization problems" in *Optimal Control of ODEs and DAEs*, Berlin, Germany: Walter de Gruyter GmbH, 2012, pp. 75-95.
- [46] B. Shi, H. Peng, X. Wang, W. Zhong, L. Gao, and J. Fottner, "A symplectic indirect approach for a class of nonlinear optimal control problems of differential-algebraic systems," *Int. J. Robust Nonlinear Control*, vol. 31, no. 7, pp. 2712-2736, 2021.
- [47] J. Chung, and G. Hulbert, "A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized- α method," *J. Appl. Mech.*, vol. 60, no. 2, 1993, pp. 371-375.
- [48] C. L. Bottasso, D. Dopicco, and L. Trainelli, "On the optimal scaling of index three DAEs in multibody dynamics," *Multibody Syst. Dyn.*, vol. 19, no. 1-2, pp. 3-20, 2007.
- [49] W. X. Zhong, and R. L. Zhang, "Parametric variational-principles and their quadratic-programming solutions in plasticity," *Comput. Struct.*, vol. 30, no. 4, pp. 887-896, 1988.
- [50] M. Kojima, N. Megido, and S. Mizuno, "A primal-dual infeasible interior point algorithm for linear programming," *Math. Program.*, vol. 61, no. 3, pp. 263-280, Sep. 1993.
- [51] C. Kanzow, "Some noninterior continuation methods for linear complementarity problems," *SIAM J. Matrix Anal. Appl.*, vol. 17, no. 4, pp. 851-868, Oct. 1996.
- [52] S. C. Billups, and G. M. Katta, "Complementarity problems," *J. Comput. Appl. Math.*, vol. 124, no. 1-2, pp. 303-318, 2000.
- [53] R. Seydel, "Stability of fixed points" in *Practical Bifurcation and Stability Analysis*, 3rd ed. New York, NY, USA: Springer-Verlag, 2010, pp. 34-35.



Fei Li (Member, IEEE) received the Ph.D. degree in engineering mechanics from Dalian University of Technology, Dalian, China, in 2022.

He is currently a Postdoctoral researcher with the School of Mechanics and Aerospace Engineering, Dalian University of Technology, Dalian, China.

His current research interests include multibody dynamics and control, distributed model predictive control, and tensegrity continuum robot design and control.



Hao Yang received the B.E. degree in ocean technology from Dalian University of Technology, Dalian, China, in 2020. He is currently working toward the Ph.D. degree in engineering mechanics with Dalian University of Technology, Dalian, China.

His current research interests include design, modeling and control of continuum robots.



Guoying Gu (Senior Member, IEEE) received the B.E. degree (with honors) in electronic science and technology, and the Ph.D. degree (with honors) in mechatronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2006 and 2012, respectively.

He was a Humboldt Fellow with the University of Oldenburg, Oldenburg, Germany. He was a Visiting Scholar with the Massachusetts Institute of Technology, National University of Singapore, and Concordia University. His research interests include soft robotics, bioinspired and wearable robots, smart materials sensing, actuation, and motion control. He is the author or co-author of more than 120 publications, which have appeared in *Science Robotics*, *Nature Biomedical Engineering*, *Science Advances*, *IEEE Transactions on Robotics*, etc.

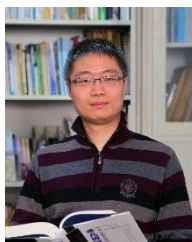
Dr. Gu was the recipient of the National Science Fund for Distinguished Young Scholars and the XPLORER PRIZE. He is an Associate Editor of *Soft Robotics* and was an Associate Editor for *IEEE Transactions on Robotics*.



Yongqing Wang received the B.E. degree in Mechanical design and manufacturing from Chengdu University of Science and Technology (now Sichuan University) Sichuan, China, in 1991 and the Ph.D. degree in Mechanical and Electronic Engineering from Dalian University of Technology, Dalian, China, in 2002.

He is currently a Professor with School of Mechanical Engineering, Dalian University of Technology, and served as the Executive Director of the Intelligent Manufacturing Longcheng Laboratory.

Dr. Wang was the recipient of Changjiang Scholar Distinguished Professor. His research interests include the intelligent manufacturing and the green manufacturing. He has published more than 150 publications, and participated in winning the first prize of the National Technology Invention Award, the first prize of the National Science and Technology Progress Award, the second prize of the National Technology Invention Award, and the second prize of the National Teaching Achievement Award.



Haijun Peng (Member, IEEE) received the Ph.D. degree in dynamics and control from Dalian University of Technology, Dalian, China, in 2012.

He is currently a Professor with the School of Mechanics and Aerospace Engineering, and fixed personnel with the State Key Laboratory of Structural Analysis, Optimization and CAE Software for Industrial Equipment, Dalian University of Technology, Dalian, China.

He received the national excellent young scientists fund in 2019. He is the author or co-author of more than 100 publications. His research interests include the integration design of structure and mechanism, multibody dynamics, and intelligent control of continuum robots.