ORIGINAL ARTICLE

Open Access

Variable Curvature Modeling Method of Soft Continuum Robots with Constraints

Yuwang Liu[1,2*], Wenping Shi[1,2,3], Peng Chen[1,2,4], Liang Cheng[1,2,4], Qing Ding[1,2,4] and Zhaoyan Deng[1,2,5]

Abstract

The inherent compliance of continuum robots holds great promise in the fields of soft manipulation and safe human–robot interaction. This compliance reduces the risk of damage to the manipulated object and its surroundings. However, continuum robots possess theoretically infinite degrees of freedom, and this high flexibility usually leads to complex deformations when subjected to external forces and positional constraints. Describing these complex deformations is the main challenge in modeling continuum robots. In this study, we investigated a novel variable curvature modeling method for continuum robots, considering external forces and positional constraints. The robot configuration curve is described using the developed mechanical model, and then the robot is fitted to the curve. A ten-section continuum robot prototype with a length of 1 m was developed in order to validate the model. The feasibility and accuracy of the model were verified by the ability of the robot to reach target points and track complex trajectories with a load. This work was able to serve as a new perspective for the design analysis and motion control of continuum robots.

Keywords Continuum robots, Variable curvature modeling, Boundary conditions, Nonlinear mechanics

1 Introduction

Continuum robots have garnered significant interest in the field of robotics, owing to their high intrinsic compliance, environmental adaptability, and operational safety [1, 2]. A large number of researchers have conducted related research [3, 4], resulting in the development of various new continuum robots. Renda et al. [5, 6] designed cable-driven continuum robots. Marchese et al. [7–9] developed pneumatic continuum robots. Gu et al. [10, 11] developed dielectric elastomer-driven continuum

robots. Kim and Lin conducted research on magnetic actuation-based soft robotics [12, 13]. These continuum robots show great potential across a wide array of applications, including medical equipment, unstructured environment exploration and soft manipulation [14–16]. However, continuum robots possess a theoretically infinite number of degrees of freedom (DOF), and this high flexibility allows complex deformations of the robot in response to external forces and positional constraints. Therefore, accurately and efficiently modeling continuum robots with external forces remains a challenging task [17, 18].

Unlike traditional rigid rod robots, continuum robots achieve movement by deforming themselves. Therefore, the kinematics of continuum robots can be replaced by mechanical analysis. When solving the model, continuum robots are usually discretized into a series of points. Mathematically, a spatial curve can be determined by these positions of points. The objective is to fit the continuum robots as closely as possible to these spatial curves [19, 20]. The calculation methods for spatial

*Correspondence:
Yuwang Liu
liuyuwang@sia.cn
[1] State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[2] Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China
[3] School of Mechanical Engineering, Northeastern University, Shenyang 110016, China
[4] University of Chinese Academy of Sciences, Beijing 100049, China
[5] School of Automation and Electrical Engineering, Shenyang Ligong University, Shenyang 110016, China

curves mainly include the constant curvature approach and variable curvature approach.

The constant curvature approach is a simplified approach for modeling continuum robots, assuming that the curvature of the curve between discrete points is the same. The advantages of closed kinematics and ease of solution make this method widely used in continuum robot modeling. In order to improve the accuracy and dexterity, Jones et al. [21] developed a geometrical approach for modeling a constant curvature continuum robot, leading to a closed-form model. Freixedes et al. [22] established an optimization framework for continuum robots based on the constant curvature assumption and derived optimized structural parameters. They proposed a kinematic model to describe the deflection characteristics of the contact–assisted continuum robot and conducted the experimental verification. Webster et al. [23] unified the kinematics and differential kinematics results of single-segment and multi-segment continuous robots with constant curvature within a common coordinate system and symbol setting. Della et al. [24] analyzed the limitations of constant models and proposed an alternative state representation to solve these issues. Simulation cases were used to support the theoretical analysis. In order to simulate interactions with the environment, Schiller et al. [25] extended the constant curvature method based on energy minimization techniques, and the proposed model exhibited effective robot kinematics.

Due to constant curvature approaches do not completely match all the characteristics of continuum robots, various variable curvature approaches have been developed. Based on the Lagrangian polynomial series solution method, Ritz and Ritz-Galerkin methods, Hadi et al. [26] minimized the configuration of the continuum robot to the geometric positions of a few physical points. Singh et al. [27], using Pythagorean hodograph (PH) curves, proposed a quantitative modeling method to obtain three-dimensional reconstructions of the configurations of a continuum robot with variable curvature. Gonthina et al. [28] proposed a cross-sectional modeling method for variable curvature continuum robots based on Eulerian spiral curves. They compared the simulation results with the constant curvature method, proving that the proposed method significantly better matched various hardware configurations of the robot. The modeling approach described above provides a good description of the forward and inverse kinematics of a soft continuous robot with variable curvature, with a primary focus on the geometric description of the robot, without considering the effect of external forces. Therefore, additional visual or displacement sensors are necessary to measure the real-time robot configuration. Some scholars have considered the influence of external forces and proposed

meaningful modeling methods. Godage et al. [29] simulated the transient and steady state dynamics of a continuum robot prototype based on the lumped parameter model. Renda et al. [30] established a mechanical model for a short thick continuum robot under the influence of external forces using the Cosserat theory. The experiment verified the most typical movements of the octopus: bending, stretching and grasping. Based on the finite element method (FEM), Bieze et al. [31] obtained the kinematics of two different continuum robots with complex structural geometries. Lumped parameter models are known for reducing model complexity, albeit at the cost of accuracy. Cosserat theory and finite element method are typically computationally intensive. In summary, this study was motivated by the ongoing challenge of performing rapid simulations for continuum robot setups.

This paper proposed a novel modeling method for continuum robots based on the principles of virtual work and vector mechanics, enabling quick and accurate calculation of continuum robot configurations. First, the equilibrium equations for continuum robots were developed, and discretized using the finite difference method (FDM). Subsequently, the least squares method was applied to transform the equation solution into an optimization problem. A 10-section continuum robot driven by pneumatic artificial muscles (PAMs) was developed, and the accuracy of the model was experimentally verified.

The rest of this paper is organized as follows. Section 2 presents a variable curvature model for continuum robots, considering external forces and positional constraints. Section 3 details two sets of experiments: one involving movement to target points with a load, and the other focusing on complex trajectory tracking with a load, serving to validate the model. Finally, the conclusion is presented in Section 4.

2 Modeling
2.1 Main Model
The model is applicable to many types of continuum robots driven by different mechanisms. Because we used a pneumatic continuum robot for our experiments, we modeled the pneumatic continuum robot as an example. In general, the structure of pneumatic continuum robots had a central backbone and multiple sections [32], as shown in Figure 1(a). Each section was composed of three PAMs and a constraint disk, enabling a 2-DOF bending motion. The motion of the robot can be achieved by deformation of the backbone, leading to a bend in three-dimensional space by adjusting the lengths of the driving PAMs. Therefore, the configurations of the backbone serve as the configurations of the robot, and the robot model mainly focuses on analyzing the backbone.
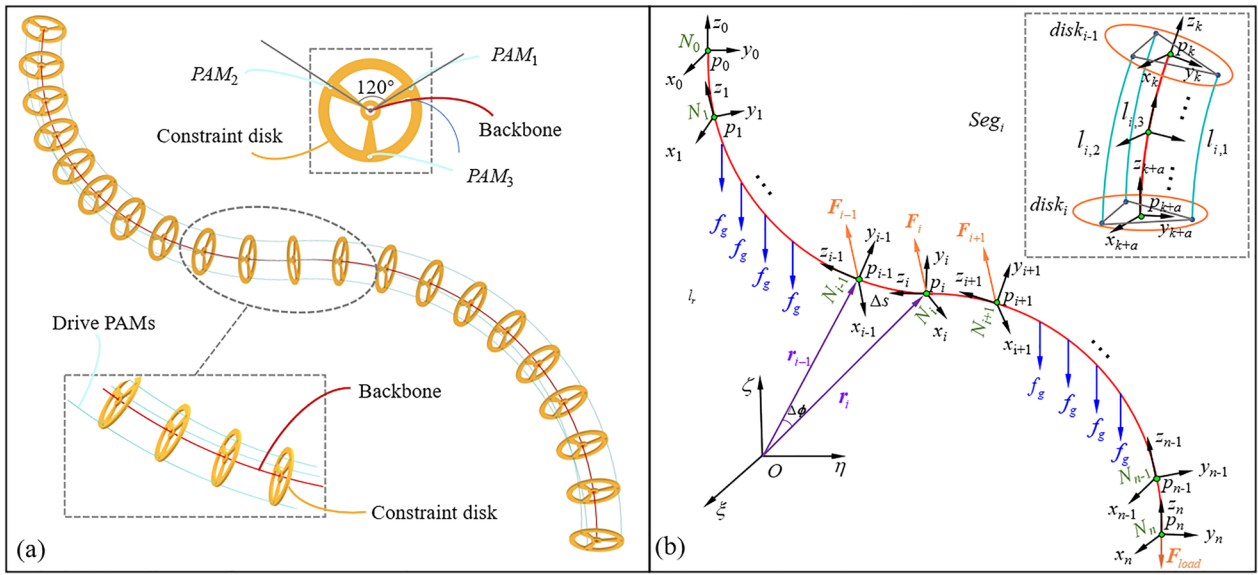
Figure 1 (**a**) Generalized structure of pneumatic continuum robots, (**b**) Geometric schematic view used to describe a continuum robot

The backbone is discrete into n elements, and N_i represents the ith nodes. There are m constraint disks, and the segment between $disk_{i-1}$ and $disk_i$ is called Seg_i, which length is Δs. A static analysis was performed on the micro element Seg_i of the robot in the inertial coordinate system $O - \xi\eta\zeta$, as shown in Figure 1(b). The spindle coordinate system of N_i is $p_i - x_i y_i z_i$. The vector of node N_i with respect to O are $\boldsymbol{r}_i$. The internal force at N_i are $\boldsymbol{F}_i$. The gravity distribution force is $\boldsymbol{f}_g$. The relative rotation angle between the sections of adjacent nodes is $\Delta\boldsymbol{\phi}$. Evidently, there are a ($a = n/m$) nodes in Seg_i. The length of PAM_j in Seg_i is $l_{i,j}$. The external load force at the endpoint of the robot is $\boldsymbol{F}_{load}$.

Suppose the projection vector of the z-axis unit vector in $O - \xi\eta\zeta$ is T. The back-bone configuration can be obtained from Eq. (1):

$$r(s) = \int_0^s \boldsymbol{T}(\delta)\mathrm{d}\delta, \tag{1}$$

where δ denotes the integral variable.

Suppose the rate of section angular displacement $\boldsymbol{\phi}$ with respect to arc coordinate s is $\boldsymbol{\omega}$:

$$\boldsymbol{\omega} = \lim_{\Delta s \to 0} \frac{\Delta\boldsymbol{\phi}}{\Delta s}. \tag{2}$$

Using the infinitesimal rotation theory of a rigid body, the relationship between $\boldsymbol{\omega}$ and quaternions (q_1, q_2, q_3, q_4) can be derived:

$$
\begin{cases}
\omega_x = 2(-q_2 \dfrac{\mathrm{d}q_1}{\mathrm{d}s} + q_1 \dfrac{\mathrm{d}q_2}{\mathrm{d}s} + q_4 \dfrac{\mathrm{d}q_3}{\mathrm{d}s} - q_3 \dfrac{\mathrm{d}q_4}{\mathrm{d}s}), \\[2mm]
\omega_y = 2(-q_3 \dfrac{\mathrm{d}q_1}{\mathrm{d}s} - q_4 \dfrac{\mathrm{d}q_2}{\mathrm{d}s} + q_1 \dfrac{\mathrm{d}q_3}{\mathrm{d}s} + q_2 \dfrac{\mathrm{d}q_4}{\mathrm{d}s}), \\[2mm]
\omega_z = 2(-q_4 \dfrac{\mathrm{d}q_1}{\mathrm{d}s} + q_3 \dfrac{\mathrm{d}q_2}{\mathrm{d}s} - q_2 \dfrac{\mathrm{d}q_3}{\mathrm{d}s} + q_1 \dfrac{\mathrm{d}q_4}{\mathrm{d}s}).
\end{cases} \tag{3}
$$

Writing Eq. (3) in matrix form,

$$\boldsymbol{\omega} = \boldsymbol{\Gamma} \boldsymbol{q}', \tag{4}$$

where $\boldsymbol{q}' = \frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}s}$, and $\boldsymbol{\Gamma}$ is denoted as

$$
\boldsymbol{\Gamma} = 2 \begin{bmatrix} -q_2 & q_1 & q_4 & -q_3 \\ -q_3 & -q_4 & q_1 & q_2 \\ -q_4 & q_3 & -q_2 & q_1 \end{bmatrix}. \tag{5}
$$

The total energy E_t of the backbone includes elastic strain energy E_e and external force potential energy E_p. When the robot is in balance,

$$\delta E_t = \delta E_e + \delta E_p = 0. \tag{6}$$

The elastic strain energy E_e of the backbone is

$$
E_e = \frac{\int_0^L \left[k_x \left(\omega_x - \omega_x^0 \right)^2 + k_y \left(\omega_y - \omega_y^0 \right)^2 + k_z \left(\omega_z - \omega_z^0 \right)^2 \right] \mathrm{d}s}{2}, \tag{7}
$$

Liu *et al. Chinese Journal of Mechanical Engineering*   (2023) 36:148

Page 4 of 11

where L denotes the length of the robot, and (k_x, k_y, k_z) represents the bending stiffness around the coordinate axis, and $(\omega_\xi^0, \omega_\eta^0, \omega_\zeta^0)$ represents the initial state of $\boldsymbol{\omega}$

$$\begin{cases} k_x = E\dfrac{\pi d^4}{64}, \\ k_y = E\dfrac{\pi d^4}{64}, \\ k_z = G\dfrac{\pi d^4}{32}. \end{cases} \tag{8}$$

The variation of Eq. (7) is

$$\delta E_e = \int_0^L \Big[k_x(\omega_x - \omega_x^0)\delta\omega_x + k_y(\omega_y - \omega_y^0)\delta\omega_y + \\ k_z(\omega_z - \omega_z^0)\delta\omega_z \Big] \mathrm{d}s. \tag{9}$$

Writing Eq. (9) in matrix form,

$$\delta E_e = \int_0^L \left\{ \Big[\boldsymbol{K}(\boldsymbol{\omega} - \boldsymbol{\omega}^0) \Big]^{\mathrm{T}} \delta\boldsymbol{\omega} \right\} \mathrm{d}s, \tag{10}$$

where $\boldsymbol{\omega}^0 = \big(\omega_x^0 \ \omega_y^0 \ \omega_z^0 \big)^{\mathrm{T}}, \boldsymbol{K} = \mathrm{diag}\big(k_x \ k_y \ k_z \big)$.
The variation of Eq. (4) is

$$\delta\boldsymbol{\omega} = \boldsymbol{\Gamma}\delta\boldsymbol{q}' - \boldsymbol{\Gamma}'\delta\boldsymbol{q}. \tag{11}$$

Substituting Eq. (11) into Eq. (10), we obtain

$$\delta E_e = \int_0^L \left\{ \Big[\boldsymbol{K}(\boldsymbol{\omega} - \boldsymbol{\omega}^0) \Big]^{\mathrm{T}} \big(\boldsymbol{\Gamma}\delta\boldsymbol{q}' - \boldsymbol{\Gamma}'\delta\boldsymbol{q} \big) \right\} \mathrm{d}s. \tag{12}$$

The external force potential energy E_p can be expressed by the internal force $\boldsymbol{F}$ as

$$\delta E_p = -\delta \int_0^L \boldsymbol{F} \cdot \boldsymbol{T} \, \mathrm{d}s. \tag{13}$$

In the $O - \xi\eta\zeta$, the force balance of Seg_i is

$$\boldsymbol{F}_i - \boldsymbol{F}_{i-1} + \boldsymbol{f}_g \Delta s = \boldsymbol{0}. \tag{14}$$

Dividing the sides of Eq. (14) by Δs, and considering the condition $\Delta s \to 0$, we obtain

$$\frac{\mathrm{d}\boldsymbol{F}}{\mathrm{d}s} + \boldsymbol{f}_g = \boldsymbol{0}. \tag{15}$$

In the spindle coordinate system $p - xyz$, Eq. (15) needs to be rewritten as

$$\frac{\mathrm{d}\boldsymbol{F}}{\mathrm{d}s} + \boldsymbol{\omega} \times \boldsymbol{F} + \boldsymbol{f}_g = \boldsymbol{0}. \tag{16}$$

2.2 Discretization of the Equations

The quaternions at node N_i is denoted as $\boldsymbol{q}_i$, where $\boldsymbol{q}_i = \begin{bmatrix} q_{1,i} & q_{2,i} & q_{3,i} & q_{4,i} \end{bmatrix}^{\mathrm{T}}$. By performing linear interpolation in Seg_i, the quaternions and their derivatives in Seg_i are as follows:

$$\begin{cases} \bar{q}_{k,i} = \dfrac{q_{k,i-1} + q_{k,i}}{2}, \\ q'_{k,i} = \dfrac{q_{k,i} - q_{k,i-1}}{\Delta s}, \end{cases} (k = 1, 2, 3, 4; i = 1, 2, ..., n). \tag{17}$$

The quaternions at nodes N_{i-1} and N_i are combined into an 8-order array, denoted as

$$\boldsymbol{q}_{i-1,i} = \big(\boldsymbol{q}_{i-1}^{\mathrm{T}} \ \boldsymbol{q}_i^{\mathrm{T}} \big)^{\mathrm{T}}. \tag{18}$$

Then, Eq. (17) can be expressed as

$$\bar{\boldsymbol{q}}_i = \boldsymbol{\Psi}_4 \boldsymbol{q}_{i-1,i}, \boldsymbol{q}'_i = \boldsymbol{\Phi}_4 \boldsymbol{q}_{i-1,i}, \tag{19}$$

where $\boldsymbol{\Psi}_4$ and $\boldsymbol{\Phi}_4$ are 4×8 matrices composed of a 4-order unit matrix $\boldsymbol{E}_4$

$$\boldsymbol{\Psi}_4 = \frac{1}{2}\big(\boldsymbol{E}_4 \ \boldsymbol{E}_4 \big), \ \boldsymbol{\Phi}_4 = \frac{1}{\Delta s}\big(-\boldsymbol{E}_4 \ \boldsymbol{E}_4 \big). \tag{20}$$

The average and derivative of $\boldsymbol{\Gamma}$ in Seg_i are denoted as $\boldsymbol{\Gamma}_i$ and $\boldsymbol{\Gamma}'_i$, respectively.

$$\begin{cases} \overline{\boldsymbol{\Gamma}}_i(\bar{\boldsymbol{q}}_i) = \dfrac{1}{2}\big[\boldsymbol{\Gamma}_{i-1}(\boldsymbol{q}_{i-1}) + \boldsymbol{\Gamma}_i(\boldsymbol{q}_i) \big], \\ \boldsymbol{\Gamma}'(\boldsymbol{q}'_i) = \dfrac{1}{\Delta s}\big[\boldsymbol{\Gamma}_{i-1}(\boldsymbol{q}_{i-1}) - \boldsymbol{\Gamma}_i(\boldsymbol{q}_i) \big], \end{cases} \tag{21}$$

where

$$\boldsymbol{\Gamma}_i(\boldsymbol{q}_i) = 2 \begin{bmatrix} -q_{2,i} & q_{1,i} & q_{4,i} & -q_{3,i} \\ -q_{3,i} & -q_{4,i} & q_{1,i} & q_{2,i} \\ -q_{4,i} & q_{3,i} & -q_{2,i} & q_{1,i} \end{bmatrix}. \tag{22}$$

The average and derivative of $\boldsymbol{\omega}$ in Seg_i are denoted as $\bar{\boldsymbol{\omega}}_i$ and $\boldsymbol{\omega}'_i$, respectively:

$$\bar{\boldsymbol{\omega}}_i = \overline{\boldsymbol{\Gamma}}_i(\bar{\boldsymbol{q}}_i)\boldsymbol{q}'_i, \delta\bar{\boldsymbol{\omega}}_i = \overline{\boldsymbol{\Gamma}}_i(\bar{\boldsymbol{q}}_i)\delta\boldsymbol{q}'_i - \boldsymbol{\Gamma}'(\boldsymbol{q}'_i)\delta\bar{\boldsymbol{q}}_i. \tag{23}$$

Substituting Eq. (19) into Eq. (23), we obtain

$$\begin{cases} \bar{\boldsymbol{\omega}}_i = \overline{\boldsymbol{\Gamma}}_i \boldsymbol{\Phi}_4 \boldsymbol{q}_{i-1,i}, \\ \delta\bar{\boldsymbol{\omega}}_i = (\overline{\boldsymbol{\Gamma}}_i \boldsymbol{\Phi}_4 - \boldsymbol{\Gamma}'_i \boldsymbol{\Psi}_4)\delta\boldsymbol{q}_{i-1,i}. \end{cases} \tag{24}$$

Eq. (12) can be discretized as

$$\delta E_e = \sum_{i=1}^{n} \left[\left(\boldsymbol{q}_{i-1,i}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\Phi}_4^{\mathrm{T}} \overline{\boldsymbol{\Gamma}}_i^{\mathrm{T}} - \boldsymbol{\omega}_i^{0\mathrm{T}} \right) \left(\overline{\boldsymbol{\Gamma}}_i \boldsymbol{\Phi}_4 - \boldsymbol{\Gamma}_i' \boldsymbol{\Psi}_4 \right) \right] \delta \boldsymbol{q}_{i-1,i}. \tag{25}$$

Eq. (25) can be simplified as

$$\delta E_e = \sum_{i=1}^{n} (\boldsymbol{q}_{i-1,i}^{\mathrm{T}} \boldsymbol{A}_i - \boldsymbol{B}_i) \delta \boldsymbol{q}_{i-1,i}, \tag{26}$$

where

$$\begin{cases} \boldsymbol{A}_i = \boldsymbol{K} \boldsymbol{\Phi}_4^{\mathrm{T}} \overline{\boldsymbol{\Gamma}}_i^{\mathrm{T}} (\overline{\boldsymbol{\Gamma}}_i \boldsymbol{\Phi}_4 - \boldsymbol{\Gamma}_i' \boldsymbol{\Psi}_4), \\ \boldsymbol{B}_i = \boldsymbol{\omega}_i^{0\mathrm{T}} (\overline{\boldsymbol{\Gamma}}_i \boldsymbol{\Phi}_4 - \boldsymbol{\Gamma}_i' \boldsymbol{\Psi}_4), \end{cases} \tag{27}$$

where $\boldsymbol{A}_i$ is arranged diagonally, and each matrix is moved 4 rows and 4 columns to the upper left, with overlapping elements added to form matrix $\boldsymbol{A}$. Similarly, $\boldsymbol{B}_i$ is arranged horizontally, and each matrix is moved four columns to the left, with overlapping elements added to form matrix $\boldsymbol{B}$. Eq. (25) can be expressed as

$$\delta E_e =, (\boldsymbol{q}^{\mathrm{T}} \boldsymbol{A} - \boldsymbol{B}) \delta \boldsymbol{q}, \tag{28}$$

where $\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_0^{\mathrm{T}} & \boldsymbol{q}_1^{\mathrm{T}} & \dots & \boldsymbol{q}_n^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$.

The projection of $\boldsymbol{T}$ in $O - \xi\eta\zeta$ can be expressed by quaternions as

$$\boldsymbol{T} = \begin{bmatrix} 2(q_2 q_4 + q_1 q_3) & 2(q_3 q_4 - q_1 q_2) & 2(q_1^2 + q_4^2) - 1 \end{bmatrix}^{\mathrm{T}}, \tag{29}$$

where $\boldsymbol{T}_i$ can be discretized as

$$\boldsymbol{T}_i = \frac{1}{3} \begin{bmatrix} (2q_{2,i-1} + q_{2,i})q_{4,i-1} + (2q_{2,i} + q_{2,i-1})q_{4,i} + \\ (2q_{3,i-1} + q_{3,i})q_{4,i-1} + (2q_{3,i} + q_{3,i-1})q_{4,i} - \\ \sum_{k \in \{1,4\}} (q_{k,i-1}^2 + q_{k,i-1} q_{k,i} + q_{k,i}^2) - \end{bmatrix}$$

$$\begin{bmatrix} (2q_{1,i-1} + q_{1,i})q_{3,i-1} + (2q_{1,i} + q_{1,i-1})q_{3,i} \\ (2q_{1,i-1} + q_{1,i})q_{2,i-1} - (2q_{1,i} + q_{1,i-1})q_{2,i} \\ \sum_{k \in \{2,3\}} (q_{k,i-1}^2 + q_{k,i-1} q_{k,i} + q_{k,i}^2) \end{bmatrix}. \tag{30}$$

Then, $\boldsymbol{T}_i$ can be expressed by $\boldsymbol{S}_i$ as

$$\boldsymbol{T}_i = \frac{1}{3} \boldsymbol{S}_i \boldsymbol{q}_{i-1,i}, \tag{31}$$

where

$$\boldsymbol{S}_i^{(1)} = \begin{bmatrix} 2q_{3,i-1} & 2q_{4,i-1} & q_{1,i} & q_{2,i} \\ -2q_{2,i-1} & -q_{1,i} & 2q_{4,i-1} & q_{3,i} \\ q_{1,i-1} + q_{1,i} & -(q_{2,i-1} + q_{2,i}) & -(q_{3,i-1} + q_{3,i}) & q_{4,i-1} + q_{4,i} \end{bmatrix}, \tag{32}$$

$$\boldsymbol{S}_i^{(2)} = \begin{bmatrix} 2q_{3,i} & 2q_{4,i} & q_{1,i-1} & q_{2,i-1} \\ -2q_{2,i} & -q_{1,i-1} & 2q_{4,i} & q_{3,i-1} \\ q_{1,i} & q_{2,i} & q_{3,i} & q_{4,i} \end{bmatrix}, \tag{33}$$

$$\boldsymbol{S}_i = \begin{bmatrix} \boldsymbol{S}_i^{(1)} & \boldsymbol{S}_i^{(2)} \end{bmatrix}. \tag{34}$$

The average and derivative of $\boldsymbol{F}$ in Seg_i are denoted as $\overline{\boldsymbol{F}}_i$ and $\boldsymbol{F}_i'$, respectively.

$$\overline{\boldsymbol{F}}_i = \frac{\boldsymbol{F}_{i-1} + \boldsymbol{F}_i}{2} \quad \boldsymbol{F}_i' = \frac{\boldsymbol{F}_i' - \boldsymbol{F}_{i-1}}{\Delta s}, \tag{35}$$

where $\boldsymbol{F}_i = \begin{bmatrix} F_{x,i} & F_{y,i} & F_{z,i} \end{bmatrix}^{\mathrm{T}}$.

The internal force $\boldsymbol{F}_{i-1}$ and $\boldsymbol{F}_i$ at nodes N_{i-1} and N_i are combined into a 6-order array, which is denoted as

$$\boldsymbol{F}_{i-1,i} = \begin{pmatrix} \boldsymbol{F}_{i-1}^{\mathrm{T}} & \boldsymbol{F}_i^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}}. \tag{36}$$

Then, Eq. (35) can be expressed as

$$\begin{cases} \overline{\boldsymbol{F}}_i = \boldsymbol{\Psi}_3 \boldsymbol{F}_{i-1,i}, \\ \boldsymbol{F}_i' = \boldsymbol{\Phi}_3 \boldsymbol{F}_{i-1,i}, \end{cases} \tag{37}$$

where $\boldsymbol{\Psi}_3$ and $\boldsymbol{\Phi}_3$ are 3×6 matrices composed of a 3-order unit matrix $\boldsymbol{E}_3$:

$$\begin{cases} \boldsymbol{\Psi}_3 = \frac{1}{2} \begin{pmatrix} \boldsymbol{E}_3 & \boldsymbol{E}_3 \end{pmatrix}, \\ \boldsymbol{\Phi}_3 = \frac{1}{\Delta s} \begin{pmatrix} -\boldsymbol{E}_3 & \boldsymbol{E}_3 \end{pmatrix}. \end{cases} \tag{38}$$

Eq. (13) can be discretized as

$$\delta E_p = -\frac{1}{3} \sum_{i=1}^{n} \boldsymbol{F}_{i-1,i}^{\mathrm{T}} \boldsymbol{\Psi}_3^{\mathrm{T}} \boldsymbol{S}_i \delta \boldsymbol{q}_{i-1,i}. \tag{39}$$

Let

$$\boldsymbol{U}_i = \boldsymbol{\Psi}_3^{\mathrm{T}} \boldsymbol{S}_i. \tag{40}$$

Liu *et al. Chinese Journal of Mechanical Engineering*   (2023) 36:148

Page 6 of 11

Matrices U_i are arranged horizontally, where each matrix is moved by four columns to the left to form matrix U. Then, Eq. (39) can be abbreviated as

$$\delta E_p = -F^{\mathrm{T}} U \delta q, \tag{41}$$

where $F = \begin{bmatrix} F_0^{\mathrm{T}} & F_1^{\mathrm{T}} & ... & F_n^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$.

Substituting Eqs. (28) and (41) into Eq. (6), the variation of E_t is as

$$\delta E_t = (q^{\mathrm{T}} A - B - F^{\mathrm{T}} U) \delta q. \tag{42}$$

Eq. (43) can be obtained from Eq. (6):

$$q^{\mathrm{T}} A - B - F^{\mathrm{T}} U = 0. \tag{43}$$

Eq. (16) can be discretized as

$$\begin{cases} \frac{F_{x,i} - F_{x,i-1}}{\Delta s} + \omega_{y,i} F_{z,i} - \omega_{z,i} F_{y,i} = 0, \\ \frac{F_{y,i} - F_{y,i-1}}{\Delta s} + \omega_{z,i} F_{x,i} - \omega_{x,i} F_{z,i} = 0, \qquad \forall i = 1 \sim n. \\ \frac{F_{z,i} - F_{z,i-1}}{\Delta s} + \omega_{x,i} F_{y,i} - \omega_{y,i} F_{x,i} + f_g = 0, \end{cases} \tag{44}$$

Eqs. (43) and (44) are the discrete balance equations.

2.3 Boundary Conditions

According to Eq. (1), the coordinate of any point on the backbone is

$$r(s) = \begin{bmatrix} \xi(s) \\ \eta(s) \\ \zeta(s) \end{bmatrix} = \begin{bmatrix} 2 \int_0^s (q_2(\sigma) q_4(\sigma) + q_1(\sigma) q_3(\sigma)) \, \mathrm{d}\sigma \\ 2 \int_0^s (q_3(\sigma) q_4(\sigma) - q_1(\sigma) q_2(\sigma)) \, \mathrm{d}\sigma \\ \int_0^s (q_1^2(\sigma) - q_2^2(\sigma) - q_3^2(\sigma) + q_4^2(\sigma)) \, \mathrm{d}\sigma \end{bmatrix}, \tag{45}$$

where σ denotes the integral variable.

Eq. (45) can be discretized by Eq. (30) as

$$\begin{cases} \xi_i = \frac{\Delta s}{3} ((2q_{2,i-1} + q_{2,i}) q_{4,i-1} + (2q_{2,i} + q_{2,i-1}) q_{4,i} \\ \quad + (2q_{1,i-1} + q_{1,i}) q_{3,i-1} + (2q_{1,i} + q_{1,i-1}) q_{3,i}) + \xi_{i-1}, \\ \eta_i = \frac{\Delta s}{3} ((2q_{3,i-1} + q_{3,i}) q_{4,i-1} + (2q_{3,i} + q_{3,i-1}) q_{4,i} \\ \quad - (2q_{1,i-1} + q_{1,i}) q_{2,i-1} - (2q_{1,i} + q_{1,i-1}) q_{2,i}) + \eta_{i-1}, \\ \zeta_i = \frac{\Delta s}{3} (q_{1,i-1}^2 + q_{1,i-1} q_{1,i} + q_{1,i}^2 - q_{2,i-1}^2 - q_{2,i-1} q_{2,i} - q_{2,i}^2 \\ \quad - q_{3,i-1}^2 - q_{3,i-1} q_{3,i} - q_{3,i}^2 + q_{4,i-1}^2 + q_{4,i-1} q_{4,i} + q_{4,i}^2) + \zeta_{i-1}. \end{cases} \tag{46}$$

It is typically necessary to control the endpoint's movement to a specified position when a continuum robot is in operation, so the position of the endpoint needs to be limited. Assuming the inclusion point of the robot is at the origin *O*, we obtain:

$$\begin{cases} \xi_n - P_{n,\xi} = 0, \\ \eta_n - P_{n,\eta} = 0, \\ \zeta_n - P_{n,\zeta} = 0, \end{cases} \tag{47}$$

where $(P_{n,\xi}, P_{n,\eta}, P_{n,\zeta})$ is the desired coordinate of the endpoint.

In addition, the poses of the robot at the initial and end points also need to be constrained:

$$\begin{cases} q_{1,0} - Q_{1,0} = 0, \\ q_{2,0} - Q_{2,0} = 0, \\ q_{3,0} - Q_{3,0} = 0, \\ q_{4,0} - Q_{4,0} = 0, \end{cases} \begin{cases} q_{1,n} - Q_{1,n} = 0, \\ q_{2,n} - Q_{2,n} = 0, \\ q_{3,n} - Q_{3,n} = 0, \\ q_{4,n} - Q_{4,n} = 0. \end{cases} \tag{48}$$

where $(Q_{1,0}, Q_{2,0}, Q_{3,0}, Q_{4,0})$ and $(Q_{1,n}, Q_{2,n}, Q_{3,n}, Q_{4,n})$ represent the desired quaternions of the initial and end points, respectively.

When the robot has a load, it is also necessary to add an endpoint force boundary condition

$$F_n - F_{load} = 0. \tag{49}$$

Eqs. (47)–(49) constitute the boundary conditions of the continuous robot model.

2.4 Constraint Model

The continuum robot model, composed of Eqs. (43), (44), (47), (48) and (49), can be expressed as

$$\begin{cases} h_1 = q^{\mathrm{T}} A - B - F^{\mathrm{T}} U = 0, \\ h_2 = \Phi_3 F_{i-1,i} + f_g = 0, \forall i \in 1 \sim n, \\ h_3 = r_n - P_n = 0, \\ h_4 = q_0 - Q_0 = 0, \\ h_5 = q_n - Q_n = 0, \\ h_6 = F_n - F_{load} = 0. \end{cases} \tag{50}$$

By using the least square method, Eq. (50) can be transformed into an optimization problem:

$$\begin{cases} \min H(X), \\ H = \frac{1}{2} \sum_{i=1}^u h_i^2(X), \end{cases} \tag{51}$$

where *u* is the dimension of *h*,

$$X = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & F_x & F_y & F_z \end{bmatrix}^{\mathrm{T}}.$$

The Particle Swarm Optimization (PSO) algorithm and the Levenberg-Marquardt (LM) algorithm are used to solve Eq. (51). The PSO algorithm exhibits strong global optimization capabilities but weaker local optimization abilities, while the LM algorithm exhibits the opposite characteristics. Therefore, our approach initially involves using the PSO algorithm to determine an appropriate iteration initial value, and the value is subsequently fed into the LM algorithm to optimize the solution. The

Liu *et al. Chinese Journal of Mechanical Engineering*   (2023) 36:148

Page 7 of 11

process of finding the initial iteration value involves two steps: the first step utilizes the PSO algorithm, and the second step employs the gradient descent method to modify the particle velocity term in PSO algorithm. This two-step process allows for the acquisition of a more suitable initial iteration value. The detailed procedure is shown in Algorithm 1.

$t = 1$; Generated random initial population xp^1, vp^1.

$X \leftarrow InitValue(xp^1, vp^1)$

$X \leftarrow Optimization(X)$

for $t = 1 : t_{max}$

$w^{t+1} = |\sigma c(X) - w^t|$

Update $f(X), F(X)$

$X \leftarrow Optimization(X)$

end for

function $X \leftarrow InitValue(xp^1, vp^1)$

for $k = 1 : Ite_1 + Ite_2$

$vp_i^{k+1} = \chi(\lambda^k vp_i^k + c_1 r_{i1}^k (pb_i^k - xp_i^k) + c_2 r_{i2}^k (gb^k - xp_i^k))$

$xp_i^{k+1} = xp_i^k + vp_i^{k+1}$

$pb_i^k \leftarrow$ the optimal solution of the i particle

if $k \in [1, Ite_1]$

$gb^k \leftarrow$ the optimal solution in all particles

end if

if $k \in [Ite_1 + 1, Ite_1 + Ite_2]$

$gb^k \leftarrow SearchStep(gb^k) + gb^k$

end if

$X \leftarrow gb^k$

end for

end function

function $d(X) \leftarrow SearchStep(X)$

$d(X) = -[J(X)^T J(X) + \mu I]^{-1} \nabla H(X)$

while $H(X + \beta^m d(X)) < H(X) + \alpha \beta^m \nabla H(X) d(X)$

$m = m + 1$

end

$d(X) = d(X) + \beta^m d(X)$

function $X \leftarrow Optimization(X)$

while $H(X) > \varepsilon_1 \&\& |\nabla H(X)| > \varepsilon_2$

$d(X) \leftarrow SearchStep(X)$

$\rho = \dfrac{H(X + d(X)) - H(X)}{\nabla H(X)^T d(X) + \frac{1}{2} d(X) \nabla^2 H(X) d(X)}$

$X = X + d(X)$

if $\rho < 0.45$ $\mu = \mu q$ end if

if $\rho > 0.75$ $\mu = \mu / q$ end if

if $0.45 \le \rho \le 0.75$ $\mu = \mu$ end if

end while

end function

Algorithm 1

3 Results

3.1 Experimental Preparation

A ten-section prototype of a continuum robot, driven by PAMs, was developed, as shown in Figure 2(a). The structure of the prototype is shown in Figure 2(b). The backbone is an elastic rod, and each section is driven by three PAMs. By changing the length of the PAMs, which can be achieved by inflating and deflating [33, 34], the robot can be controlled to bend in a three-dimensional space. The control system is composed of an upper computer (PC), a lower computer (STM 32), relays and solenoid valves. Communication between the upper computer and the lower computer occurred via Bluetooth. Commands are transmitted from the upper computer to the lower computer, and the opening and closing times of the solenoid valve are controlled by the relay to control the durations of inflation and deflation, thereby adjusting the length of the PAMs.

The analysis of the robot was performed in four main spaces: the actuator space, the joint space, the configuration space and the task space. For the pneumatic continuum robot, the actuator space refers to the air pressure of the PAMs, the joint space refers to the length of the PAMs, the configuration space is determined by the discrete point quaternions, and the task space refers to the coordinates of the robot endpoint. The mapping from joint space to task space is known as forward kinematics (f_{for}) and the reverse mapping is inverse kinematics (f_{inv}), as shown in Figure 2(c). By adjusting the air pressure of the PAMs, the robot can be controlled to bend in space with a load, as shown in Figure 2(d). The length of the PAMs is proportional to the internal air pressure, but precise control of the air pressure is challenging. When the inflation velocity is fixed, the length of the PAMs is proportional to the inflation/deflation time. Therefore, we control the length of the PAMs by controlling the inflation/deflation time, which can be achieved by controlling the switching time of the solenoid valves. The relationship between these variables is shown in Figure 3.

Unlike traditional link robots, the model of a continuum robot typically yields numerical solutions rather than analytical solutions. When solving the continuum robot model, the difference format and optimization algorithm will inevitably introduce computational errors, and the number of discrete elements usually has a significant impact on these errors. We compare the computational errors of different discrete elements and observe that when the number of elements increases from 5 to 20, the computational errors gradually decrease, and when they increase from 20 to 50, the computational errors stabilize. At the same time, the computation time is directly proportional to the number of elements. When the number of elements is 20, the computation time is 5 ms, as shown in Figure 4. Therefore, we have determined that 20 is the appropriate number of discrete elements.
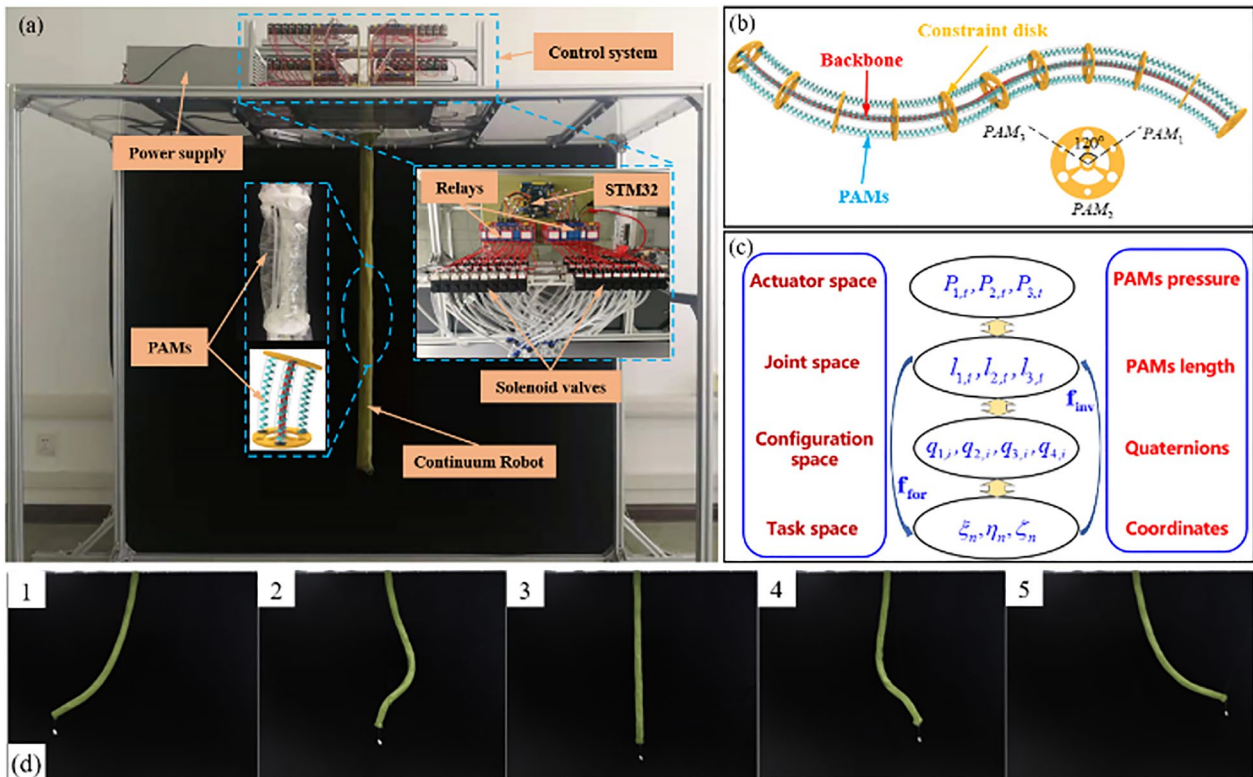
Liu *et al. Chinese Journal of Mechanical Engineering* (2023) 36:148

Page 8 of 11

Figure 2 (**a**) Experimental platform, (**b**) Schematic diagram of the prototype, (**c**) The spaces and mappings between them, (**d**) The configuration of the continuum robot with a load
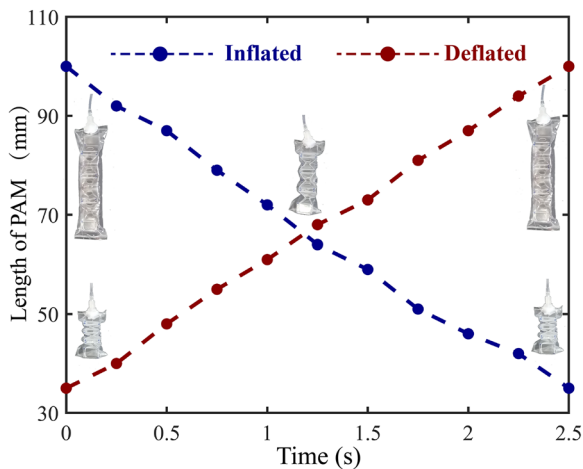

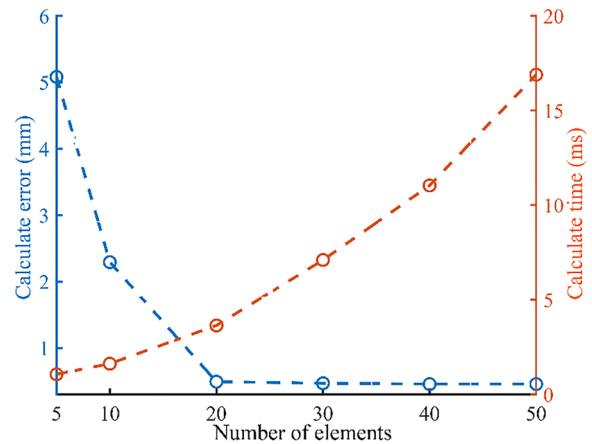**Figure 3** The relationship between the length of PAMs and the time of inflation/deflation

Figure 4 The relationship between the length of PAMs and the time of inflation/deflation

3.2 Experimental Results

In practical application scenarios, continuum robots often need to move along certain trajectories, and these trajectories could be discretized into a series of target points. Therefore, the accuracy of the model can be analyzed by calculating the error (*e*) between the robot endpoint position and the target point. The ratio of *e* to the robot length is defined as the model error. Three sets of experiments were conducted to verify the model accuracy, each involving the robot endpoint with a different load.

The robot endpoint, when not carrying any load, followed the pentagram trajectory, which was discretized into 60 target points. The movement of the robot
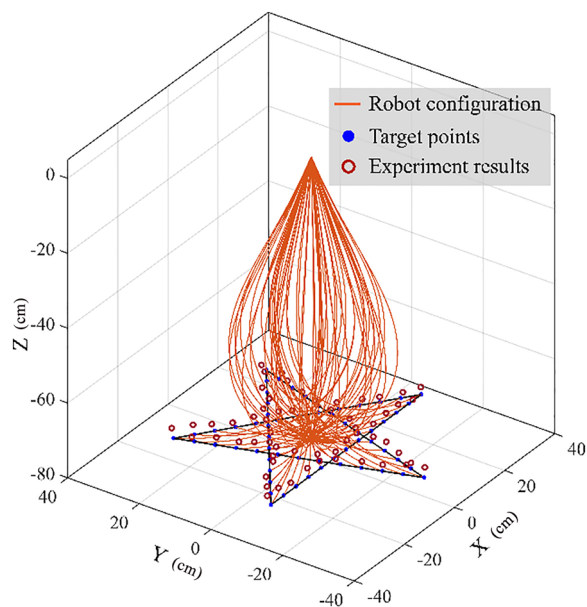
Figure 5 Comparison of simulation and experimental results for robot endpoint positions on the pentagram trajectory (without load)
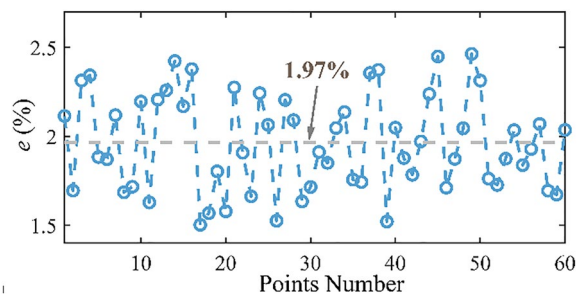
Figure 6 Errors of the robot endpoint position
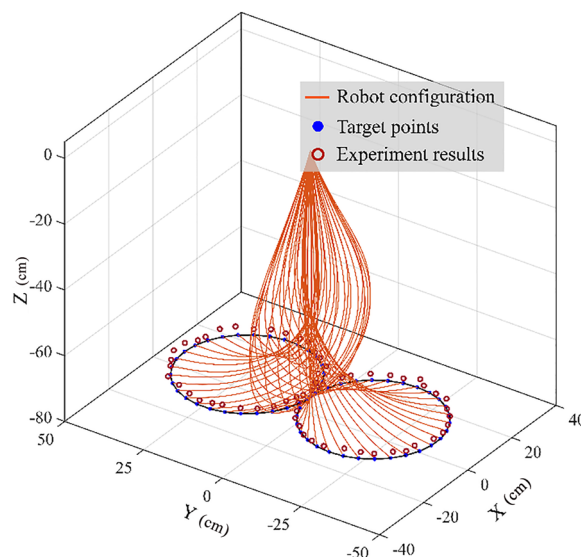

Figure 7 Comparison of simulation and experimental results for robot endpoint positions on the circular trajectory (30 g load)

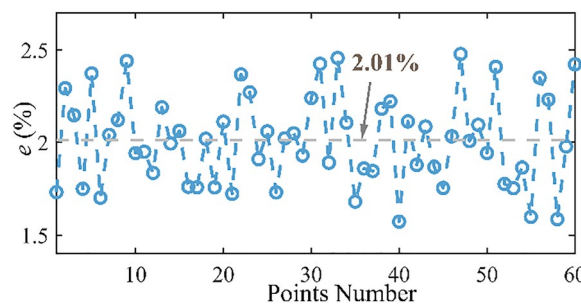**Figure 8** Errors of the robot endpoint position

endpoint was controlled based on the model. The target points and the actual endpoint positions are shown in Figure 5, and the position error of the robot endpoint is shown in Figure 6. When there is no load at the robot endpoint, the average position error is 1.97%.

The robot endpoint, with a 30 g load, followed a circular trajectory, which was also discretized into 60 target points. The comparison between the simulation and experimental results is shown in Figure 7, and the position error of the endpoint is shown in Figure 8. When the robot endpoint with a 30 g load, the average position error is 2.01%.

The robot endpoint with a 60 g load followed the space spiral trajectory, which was also discrete into 60 target points. The comparison between the simulation and experimental results is shown in Figure 9, and

the position error of the endpoint is shown in Figure 10. When the robot endpoint with a 60 g load, the average position error is 2.16%.

4 Conclusions
The conclusion of this study is summarized as follows.

(1) Based on the principle of virtual work and vector mechanics, a continuum robot modeling method is proposed, considering external loads and position constraints. The model is then numerically solved using an optimization algorithm after the equations are discretized using the finite difference technique.

(2) Three sets of experiments are conducted to validate the model. The experimental results show that with the proposed model, the configuration can be accurately predicted, and the robot can be effectively controlled to follow the desired trajectory. The average position errors of the robot endpoint when the
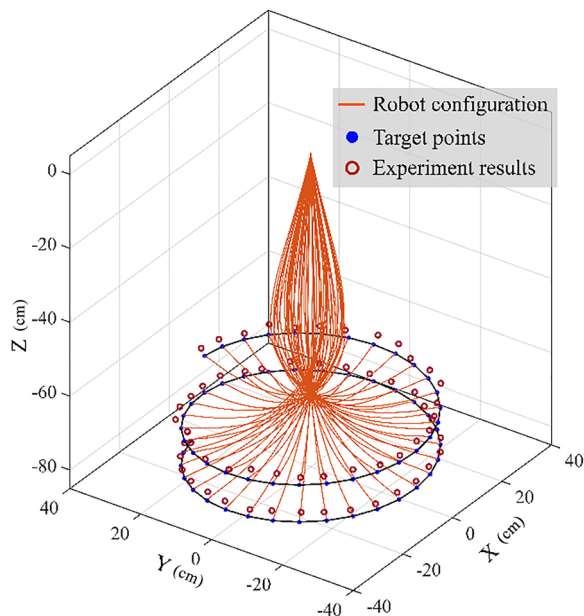
Liu *et al. Chinese Journal of Mechanical Engineering*　　(2023) 36:148

Page 10 of 11

Figure 9 Comparison of simulation and experimental results for robot endpoint positions on the space spiral trajectory (60 g load)
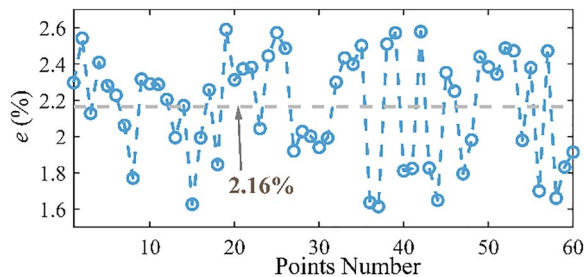

Figure 10 Errors of the robot endpoint position

loads are 0 g, 30 g and 60 g are 1.97%, 2.01% and 2.16%, indicating that the model error increases with the load. This work offers a new perspective on the design, kinematic analysis and motion planning of continuum robots.

Authors' Contributions
LYW and SWP were in charge of the whole trial. SWP and CP wrote the manuscript. SWP, CP, CL, DQ and DZY assisted with sampling and laboratory analyses. All authors read and approved the final manuscript.

Authors' Information
Yuwang Liu, born in 1982, is currently a researcher and a PhD candidate supervisor at *Shenyang Institute of Automation, Chinese Academy of Sciences, China*. His main research interests include soft robot and space robot.
Wenping Shi, born in 1995, is currently a master candidate at *Northeastern University* and received joint training at *Shenyang Institute of Automation,*

Chinese Academy of Sciences, China. His research interests include structure design and control of soft robot.
Peng Chen, born in 1996, is currently a PhD candidate at *Shenyang Institute of Automation, Chinese Academy of Sciences, China*. His research interests include modeling and control of continuum robots.
Liang Cheng, born in 1999, is currently a master candidate at *Shenyang Institute of Automation, Chinese Academy of Sciences, China*. His research interests include reconfigurable robot.
Qing Ding, born in 1998, is currently a doctor candidate at *Shenyang Institute of Automation, Chinese Academy of Sciences, China*. His research interests include biped robot.
Zhaoyan Deng, born in 1999, is currently a master candidate at *Shenyang Ligong University, China*, and received joint training at *Shenyang Institute of Automation, Chinese Academy of Sciences, China*. Her research interests include control of soft robot.

Data availability
The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Competing Interests
The authors declare no competing financial interests.

References
[1] D Rus, M T Tolley, et al. Design, fabrication and control of soft robots. *Nature*, 2015, 521(7553): 467–475.
[2] F Ahmed, M Waqas, B Shaikh, et al. Multi-material bio-inspired soft octopus robot for underwater synchronous swimming. *Journal of Bionic Engineering,* 2022, 19(5): 1229–1241.
[3] E Franco, T Ayatullah, A Sugiharto, et al. Nonlinear energy-based control of soft continuum pneumatic manipulators. *Nonlinear Dynamics*, 2021, 106(1): 229–253.
[4] A A Shabana, A E Eldeeb. Motion and shape control of soft robots and materials. *Nonlinear Dynamics*, 2021, 104(1): 165–189.
[5] F Renda, M Giorelli, M Calisti. Dynamic model of a multibending soft robot arm driven by cables. *IEEE Transactions on Robotics*, 2014, 30(5): 1109–1122.
[6] F Xu, H S Wang, K W S Au, et al. Underwater dynamic modeling for a cable-driven soft robot arm. *IEEE/ASME Transactions on Mechatronics*, 2018, 23(6): 2726–2738.
[7] A D Marchese, R K Katzschmann, D Rus. A recipe for soft fluidic elastomer robots. *Soft Robotics*, 2015, 2(1): 7–25.
[8] C Tutcu, B A Baydere, S K Talas. Quasi-static modeling of a novel growing soft-continuum robot. *International Journal of Robotics Research*, 2021, 40(1): 86–98.
[9] L A Abeach, M S Nefti, T Theodoridis, et al. A variable stiffness soft gripper using granular jamming and biologically inspired pneumatic muscles. *Journal of Bionic Engineering*, 2018, 15(2): 236–246.
[10] G Y Gu, J Zou, X H Zhao, et al. Soft wall-climbing robots. *Science Robotics*, 2018, 3(25): 1–12.
[11] G Y Gu, U Gupta, J Zhu. Modeling of viscoelastic electromechanical behavior in a soft dielectric elastomer actuator. *IEEE Transactions on Robotics*, 2017, 33(5): 1263–1271.
[12] Y Kim, G A Parada, S Liu, et al. Ferromagnetic soft continuum robots. *Science Robotics*, 2019, 4(33): 1–15.

[13] D Lin, N Jiao, Z Wang, et al. A magnetic continuum robot with multi-mode control using opposite-magnetized magnets. *IEEE Robotics and Automation Letters*, 2021, 6(2): 2485–2492.

[14] C Majidi. Soft robotics: A perspective current trends and prospects for the future. *Soft Robotics*, 2014, 1(1): 5–11.

[15] H Jiang, Z Wang, Y Jin, et al. Hierarchical control soft manipulators towards unstructured interactions. *The International Journal of Robotics Research*, 2021, 40(6): 411–434.

[16] N D Naclerio, Karsai, C M Murray, et al. Controlling subterranean forces enables a fast steerable burrowing soft robot. *Science Robotics*, 2021, 6(55): 1–11.

[17] Z Gong, X Fang, X Chen, et al. A soft manipulator for efficient delicate grasping in shallow water: modeling control and real-world experiments. *The International Journal of Robotics Research*, 2020, 40(1): 449–469.

[18] S Abbaszadeh, R Leidhold, S Hoerner. A design concept and kinematic model for a soft aquatic robot with complex bio-mimicking motion. *Journal of Bionic Engineering*, 2022, 19(1): 16–28.

[19] S B Andersson. Discretization of a continuous curve. *IEEE Transactions on Robotics*, 2008, 24(2): 456–461.

[20] J Burgner-Kahrs, D C Rucker, H Choset. Continuum robots for medical applications: A survey. *IEEE Transactions on Robotics*, 2015, 31(6): 1261–1280.

[21] B A Jones, I D Walker. Practical kinematics for real-time implementation of continuum robots. *IEEE Transactions on Robotics*, 2006, 22(6): 1087–1099.

[22] L R Freixedes, A Gao, N Liu. Design optimization of a contact aided continuum robot for endobronchial interventions based on anatomical constraints. *International Journal of Computer Assisted Radiology and Surgery*, 2019, 14(3): 1137–1146.

[23] R J Webster, B A Jones. Design and kinematic modeling of constant curvature continuum robots: A review. *International Journal of Robotics Research*, 2010, 29(13): 1661–1683.

[24] S C Della, A Bicchi, D Rus. On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control. *IEEE Robotics and Automation Letters*, 2020, 5(2): 1001–1008.

[25] L Schiller, A Seibel, J Schlattmann. A lightweight simulation model for soft robot's locomotion and its application to trajectory optimization. *IEEE Robotics and Automation Letters*, 2020, 5(2): 1199–1206.

[26] S M Hadi-Sadati, S E Naghibi, I D Walker. Control space reduction and realtime accurate modeling of continuum manipulators using Ritz and Ritz–Galerkin methods. *IEEE Robotics and Automation Letters*, 2017, 3(1): 1–7.

[27] I Singh, Y Amara, A Melingui. Modeling of continuum manipulators using pythagorean hodograph curves. *Soft Robotics*, 2018, 5(4): 425–442.

[28] P S Gonthina, A D Kapadia, I S Godage. Modeling variable curvature parallel continuum robots using Euler curves. *Proceedings of the International Conference on Robotics and Automation*, Montreal, Canada, May 20–24, 2019: 1679–1685.

[29] I S Godage, R Wirz, I D Walker. Accurate and efficient dynamics for variable-length continuum arms: A center of gravity approach. *Soft Robotics*, 2015, 2(3): 96–106.

[30] F Randa, F Boyer, J Dias. Discrete Cosserat approach for multisection soft manipulator dynamics. *IEEE Transactions on Robotics*, 2018, 34(6): 1518–1533.

[31] T M Bieze, F Largilliere, A Kruszewski. Finite element method based kinematics and closed-loop control of soft, continuum manipulators. *Soft Robotics*, 2018, 5(3): 348–364.

[32] B C Black, J Till, D C Rucker. Parallel continuum robots: Modeling, analysis, and actuation-based force sensing. *IEEE Transactions on Robotics*, 2017, 34(1): 29–47.

[33] S Li, D M Vogt, D Rus. Fluid-driven origami-inspired artificial muscles. *Proceedings of the National Academy of Sciences*, 2017, 114(50): 13132–13137.

[34] J G Lee, H Rodrigue. Origami-based vacuum pneumatic artificial muscles with large contraction ratios. *Soft Robotics*, 2019, 6(1): 109–117.