Mechanism and Machine Theory 168 (2022) 104618

Contents lists available at ScienceDirect

Mechanism and Machine Theory

journal homepage: www.elsevier.com/locate/mechmt

Variable-gain control for continuum robots based on velocity sensitivity

Xu Zhang [a,d], Yue Liu [a], David T. Branson [b], Chenghao Yang [a], Jian S. Dai [c], Rongjie Kang [a,*]

[a] Key Laboratory of Mechanism Theory and Equipment Design of the Ministry of Education, School of Mechanical Engineering, Tianjin University, China
[b] Faculty of Engineering, University of Nottingham, UK
[c] Centre for Robotics Research, King's College London, UK
[d] State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, China

ARTICLE INFO

Keywords:
Continuum robot
Kinematic model
Velocity sensitivity
Variable gain control

ABSTRACT

Kinematic control for continuum robots usually involves an inverse model to provide actuator positions according to the desired end-tip position, as well as a servo controller at the actuator level. The resulting control performance of a continuum robot is then related to its kinematic characteristics that vary at different configurations. In this paper, a kinematic model for a typical rod-driven continuum robot is presented. Following this, a kinematic parameter, velocity sensitivity, is proposed to evaluate the kinematic characteristics of the continuum robot, indicating the contribution of the individual actuators to the instant movement of the end-tip when tracking a given path. Next, a variable gain control strategy is presented to tune the servo controller with respect to the varying velocity sensitivity along the path, reducing the fluctuation of the tracking errors in real time. The simulated and experimental results show that the presented methods can effectively smooth the movement of the continuum robot over its workspace by considering the coordination between the kinematic and servo controllers.

1. Introduction

Traditional robotic control based on kinematics usually contains two levels, an inverse kinematic controller and a servo controller [1]. The inverse kinematic controller is used to convert the position error in operational space to the position error in actuation space [2,3]. Then, the servo controller is employed to eliminate the position error for the individual actuators [4]. Thus, the performance of the robotic control system highly depends on the performance of the kinematic and servo controllers.

There has been a number of works on improving the accuracy and robustness of these two types of controllers. Nowadays, the kinematic control for continuum robots basically borrows the methods used in rigid robots, which is based on a mapping between the operation space and actuation space, and calculates the compensation in the actuation space to eliminate the position error in the operation space [5,6]. Many kinematic controllers utilized the inverse Jacobian instead of nonlinear position mapping to improve the computational efficiency [5,7]. Some researchers focused on improving the accuracy of the kinematic mapping (i.e. the kinematic model). For example, Ma et al. established an accurate kinematic controller for a pneumatic soft robot through a Gaussian process [8].

* Corresponding author.
E-mail address: rjkang@tju.edu.cn (R. Kang).

https://doi.org/10.1016/j.mechmachtheory.2021.104618
Received 1 August 2021; Received in revised form 6 October 2021; Accepted 23 October 2021
Available online 4 November 2021

There were also a lot of works attempting to deal with the uncertainties in the robotic kinematics, especially for continuum robots with compliant and redundant structure. Parker et al. combined a genetic algorithm with the kinematic controller to solve the multi-solution problem for redundant robots [9]. Mao et al. applied a distance penalty function to the inverse kinematics of redundant robots and achieved obstacle avoidance by integrating neural networks [6]. As for servo controllers, most of them are based on a feedback loop to eliminate the position error of the actuator [10]. The PID (Proportional-Integral-Derivative) algorithm has been widely applied to servo controllers due to its simple structure and good robustness [11,12]. The key point in the design of PID controller is the tuning of the PID parameters. Kwok et al. applies a simulated annealing algorithms to tune the PID parameters to improve the accuracy of a PUMA 560 robot [13]. Li et al. uses neural networks to determine the PID parameters so the controller can adapt to the working conditions of the actuator [14]. However, few works have paid attention to the coordination between the kinematics of a robotic system and its servo controllers.

For a multi-jointed robotic system to track a given path, the contributions of the individual joints (i.e., actuators) to the movement of the end-tip are different and variable. This means even if the joints provide a certain change of their outputs, the resulting change of displacements on the end-tip of the robot may be different, depending on the current configuration of the robot. In other words, to make the end-tip of the robot move by a given displacement, the required displacement at the joint level may also be different. From the control point of view, the gain of the kinematic controller is variable along the tracking path. This variable kinematic gain will result in vibrations at the end-tip of the robot. For continuum robots, it will be quite difficult to suppress the vibrations due to the elastic body structure [15]. To solve this problem, the kinematic gain of the continuum robot should be identified during the tracking process, and the servo controller then used to compensate for the gain changes.

Therefore, the key point becomes how to define and identify the kinematic gain that affects the control performance of the continuum robot. Some researchers have conducted relevant studies on the evaluation of kinematic dexterity. Salisbury and Craig proposed the condition number as an index to describe the uniformity of the kinematic dexterity distribution of an articulated, multi-fingered robotic hand [16]. Yoshikawa et al. utilized the concept of manipulability to describe the degree of kinematic dexterity [17]. These works considered the kinematic dexterity or manipulability as a function of the robot configuration, yet ignored the influence of the moving direction. However, in a path tracking process, the kinematic dexterity is not only related to the current configuration of the robot, but also to the moving direction (i.e., the direction of the velocity). Yoshikawa et al. then proposed a manipulability ellipsoid to intuitively reflect the distribution of the kinematic dexterity for a robot [18]. It evaluated the kinematic dexterity in the directions of the eigenvectors in terms of the length of the ellipsoid's axes, but the evaluation of the kinematic dexterity in the other directions still requires further complex calculations. The results of the above-mentioned works were usually used to determine a reference path for the robots to avoid singularity. Cao et al. presented a manipulability-based least damped square method to achieve singularity avoidance [19]. Kim et al. optimized the size of the linkages for a robotic arm based on condition number [20]. Zhang et al. presented a path planning algorithm for mobile robots based on the manipulability to enhance the robot's flexibility [21]. What we are going to do in this paper is to identify the instant kinematic dexterity of the robot to tune the control parameters in real time, instead of off-line determination of the reference path. For this reason, we need a relatively simple and reliable way to on-line calculate the kinematic dexterity of the robot.

In this paper, a velocity based kinematic index, velocity sensitivity, is proposed to describe the instant kinematic mapping between the operational and joint space, taking into account both configuration and direction of the end-tip. Currently, most servo controllers are developed with fixed parameters that cannot compensate for the variation in robotic kinematics when tracking a path [22]. In our work, a variable gain control strategy is proposed to achieve the coordination between the kinematic and servo controllers. The idea is to identify the velocity sensitivity of the continuum robot in real time, and use this index to tune the PID parameters, so the control performance of the continuum robot along the tracking path can remain stable.

The paper is organized as follows: the kinematic and dynamic models of a continuum robot are presented in Section 2. The concept of velocity sensitivity is then proposed to describe the kinematic dexterity of continuum robots in Section 3. The variable gain control strategy based on the velocity sensitivity is developed in Section 4 and compared with the traditional kinematic control method. Section 5 validates the presented control strategy through experiments. Conclusions are eventually given in Section 6.

2. Kinematic and dynamic model of a continuum robot

This section presents the kinematic and dynamic models for a typical tentacle-like, rod-driven continuum robot. The kinematic model will be used for the analysis of the robot's velocity sensitivity in Section 3, and to establish the kinematic controller in Section 4. The dynamics model will be used as the controlled plant in the subsequent simulations.

2.1. Kinematic model of a continuum robot

A continuum robot is usually composed of several continuum modules in series. Each module includes a backbone, three or four parallelly arranged driving rods and a number of constraint disks. The constraint disks are fixed to the backbone, preventing the driving rods from buckling. The driving rods are fixed to the end disk of a module and moving through the other constraint disks freely. By controlling the displacements of the driving rods, each the continuum module can achieve 2 degrees of freedom (DOF) bending motion. In this paper, three driving rods distributed in a circle with an interval of 120° is taken as an example.

The kinematic model of the continuum robot can be represented by three spaces. The actuation space is described by the length of each driving rod $l_{i,j}(i = 1,2.j = 1,2,3)$, which make up the vector $\mathbf{l} = [l_{i,j}]^T$, where i is the number of the continuum modules, and j is the

number of driving rods in each module. The configuration space is defined by the Euler angle of parallel platform $\boldsymbol{\psi} = [\alpha_i, \beta_i]^{\mathrm{T}} (i = 1,2)$. As shown in Fig. 1, a continuum module can be kinematically divided into serially stacked parallel mechanisms. A parallel mechanism consists of a moving platform, a fixed base, three identical limbs, and a central strut connected to the platform with a universal joint. Each parallel platform has two rotational DOFs, which can be represented by two Euler angles, α and β. The Euler angle of the parallel mechanisms in the same continuum module is considered to be identical. The operational space is described the position of end-tip $\mathbf{x} = [x, y, z]^{\mathrm{T}}$.

The kinematic model of a single parallel platform has been established previously in [23]. So the kinematic relationship between the actuation and configuration space can be formulated by a Jacobian matrix as

$$\mathbf{J}_{l\psi} = \frac{\partial \mathbf{l}}{\partial \boldsymbol{\psi}} \tag{1}$$

In the same way, each continuum module can be connected in series to obtain the kinematic relationship between the operational space and the configuration space

$$\mathbf{J}_{x\psi} = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\psi}} \tag{2}$$

It is worth noting that the number of DOFs in the operational space is three if we only consider the end-tip position of the continuum robot, while the number of DOFs in the configuration space is four. So, there is no inverse matrix of $\mathbf{J}_{x\varphi}$, instead, its pseudo inverse matrix $\mathbf{J}_{x\psi}{}^{\dagger}$ is defined as

$$\mathbf{J}_{x\psi}{}^{\dagger} = \mathbf{J}_{x\psi}{}^{T} \left(\mathbf{J}_{x\psi} \mathbf{J}_{x\psi}{}^{T} \right)^{-1} \tag{3}$$

In this paper, the minimum norm solution to the inverse kinematic problem with redundant actuators is used to solve the inverse kinematic problem [24]. Thus, the mapping relation from operational space to actuation space can be obtained as

$$\dot{\mathbf{l}} = \mathbf{J}_{l\psi} \mathbf{J}_{x\psi}{}^{\dagger} \dot{\mathbf{x}} \tag{4}$$

2.2. Dynamic model of continuum robots

There have been some works on the dynamic modeling of continuum robots [25–28]. In this paper, referring to Kang's model in [29], a simplified dynamic model of continuum robot is established using the Lagrange method. Note that, the dynamic model defined in this paper is used as a controlled plant in the simulation to form a complete closed-loop system. It does not provide dynamic

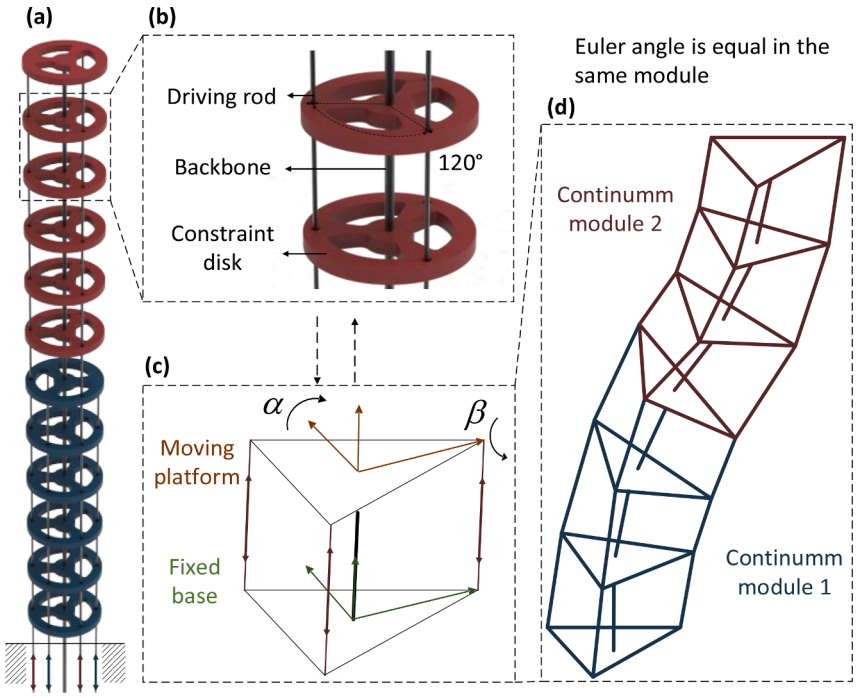


Fig. 1. The structure and kinematics of a continuum robot. (a) The structure of the continuum robot, (b) Detailed structure of a continuum module. (c) The kinematically equivalent parallel mechanism. (d) The kinematics of the continuum robot.

information to the kinematic and servo controllers presented in the following sections.

In Section 2.1, the continuum module is regarded as a series of parallel mechanisms with identical motion. Each parallel mechanism can be simplified and described by a lumped mass point with two torsional springs, as shown in Fig. 2. The mass point is in the center of the moving platform. The driving forces $f_{i,j}$ of three driving rods are equivalent to the torsion torque T_x^i, T_y^i about the α and β directions. The structural elasticity and damping of continuum module are equivalent to the torsion springs and dampers about the α and β directions.

Based on the simplified model of the parallel mechanism, the dynamic model of the continuum robot is then represented by a multi-linkage manipulator, as shown in Extended Data Fig. S1. Here the Lagrange dynamic model needs to solve the energy of each part of the system. According to the robot kinematics described in Section 2.1, the Lagrange function of the continuum robot can be obtained as

$$L = K - P_z - P_t \tag{5}$$

where K is the kinetic energy of each mass point, P_z is the gravitational potential energy of the system, P_t is the elastic potential energy of the system, i.e., the sum of the energy of each torsion spring.

Then the relationship between the driving torque matrix and Lagrange function is

$$\mathbf{T} - c\dot{\psi} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\psi}}\right) - \frac{\partial L}{\partial \boldsymbol{\psi}} \tag{6}$$

where $\mathbf{T} = \begin{bmatrix} T_x^1 & T_y^1 & T_x^2 & T_y^2 \end{bmatrix}^T$ is the matrix of driving torque, T_x^i and T_y^i is the equivalent torque of the pulling force on each driving rod acting on the equivalent mass point.

Therefore, the matrix of the driving torque is obtained as

$$\mathbf{T} = \mathbf{M}(\boldsymbol{\psi})\ddot{\psi} + \mathbf{C}(\boldsymbol{\psi}, \dot{\psi})\dot{\psi} + \mathbf{X}(\psi) \tag{7}$$

where $\boldsymbol{M}$ is inertia matrix, $\boldsymbol{C}$ is matrix of centrifugal force, Coriolis force and other related parts, $\boldsymbol{X}$ is matrix of gravity and elastic parts. (The detailed derivation process of the dynamic model and the detailed expressions of $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{X}$ are given in the appendix.) The parameters of the dynamic model used in the simulations are listed in Table 1. The detailed parameter identification and model validation can be found in [25], which is not the focus of this paper.

3. Velocity sensitivity of the continuum robot

In this section, the definition of the velocity sensitivity will be proposed based on the kinematic model of the continuum robot, and its physical meaning will be verified through a series of simulations.

3.1. Definition of the velocity sensitivity

For the continuum robot, the velocity sensitivity is defined as

$$\boldsymbol{\Lambda} = \frac{\dot{\mathbf{x}}}{\| \dot{\mathbf{i}} \|} \tag{8}$$

where $\dot{\mathbf{x}}$ is the velocity of the end-tip in the operational space, $\boldsymbol{i}$ is the velocity of the driving rods in the actuation space that can be calculated by Eq. (4). As shown in Eq. (8), the velocity sensitivity of the continuum robot, $\boldsymbol{\Lambda}$, is a vector in the same direction as the end-tip velocity$\dot{\mathbf{x}}$. Considering the unit of the velocity in the operation and actuation space are both m/s, the velocity sensitivity is dimensionless. Its 2-norm, $|\boldsymbol{\Lambda}|$, describes the kinematic gain from the actuation space to the operational space. According to Eqs. (3), (4)

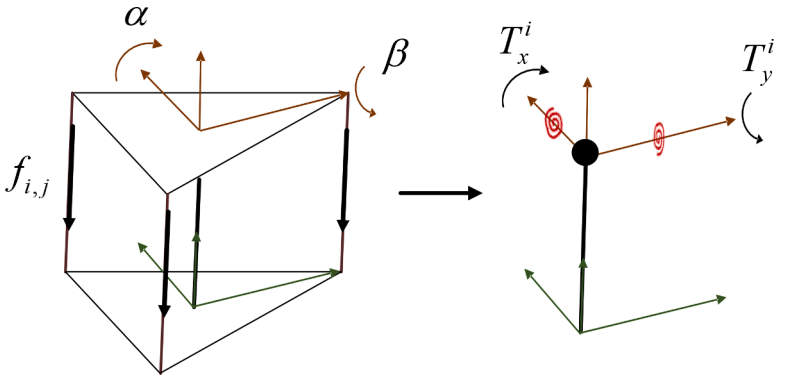


Fig. 2. Dynamics of the parallel platform.

Table 1
The parameters of the dynamics.

Property		Value
m_i	The mass of each equivalent mass point	0.05 kg
m_t	The mass of the gripper	0.06 kg
k_x	The torsion spring constant about the ***X*** direction	4 N/rad
k_y	The torsion spring constant about the ***Y*** direction	3.8 N/rad
c_x	The damping coefficient about the ***X*** direction	25 N/(rad/s)
c_y	The damping coefficient about the ***Y*** direction	20 N/(rad/s)

and (8), the velocity sensitivity, $\boldsymbol{\Lambda}$, is related to the moving direction of the end-tip in the operational space and the configuration of the continuum robot.

3.2. *Variation of the velocity sensitivity in the workspace*

According to Eq. (8), the velocity sensitivity of the continuum robot over the workspace is calculated with respect to the ***X, Y*** and ***Z*** directions, respectively, see Fig. 3. It can be found that, the velocity sensitivity in the workspace can vary greatly when the end-tip moving along different directions. For example, at the top center of the workspace, see the red area of Fig. 3(a) and (c), the 2-norm of the velocity sensitivity along the ***X*** or ***Y*** directions are very high ($\geq$120). However, in this position, the 2-norm of the velocity

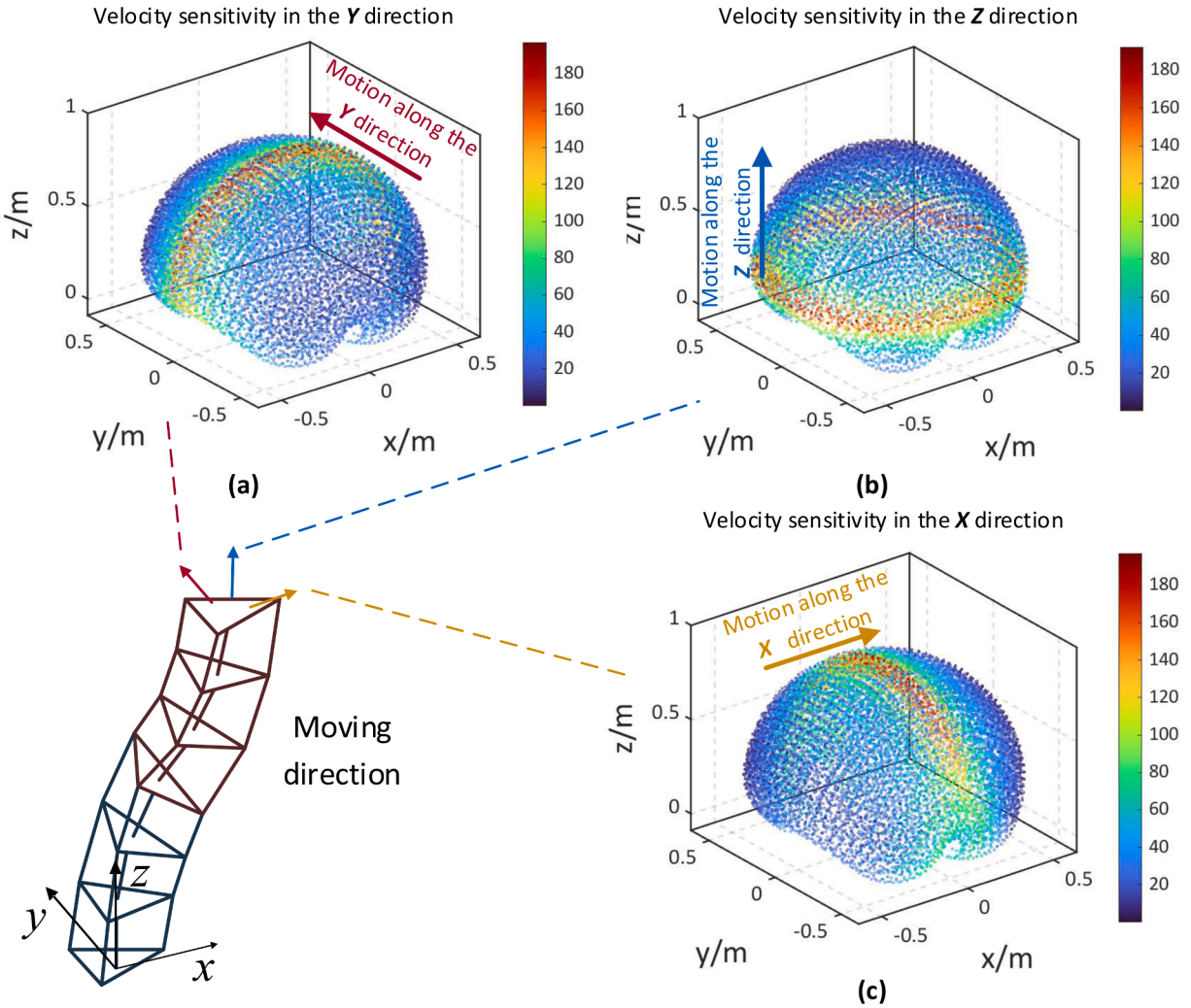


Fig. 3. The 2-norm of the velocity sensitivity, $|\boldsymbol{\Lambda}|$, over the workspace when the end-tip moving along (a) the ***Y*** direction, (b) the ***Z*** direction, (c) the ***X*** direction.

sensitivity along the $\boldsymbol{Z}$ direction is small ($\leq$40). In fact, at the top center of the workspace, there is a singularity for the robot so it lacks the ability to move in the $\boldsymbol{Z}$ direction. This shows that the robot's motion ability is not only related to its configuration, but also affected by its moving direction.

3.3. Influence of the velocity sensitivity on the control performance

In this section, a series of simulations will be conducted to investigate the influence of the velocity sensitivity on the control performance of the continuum robot.

3.3.1. Inverse kinematic control strategy

Based on the Jacobian of the continuum robot, a classic inverse kinematic control strategy is established as follows:

In this control system, $\mathbf{x}_d$ refers to the desired position of the end-tip, $\mathbf{x}_a$ refers to the actual position of the end-tip, $\boldsymbol{u}$ refers to the voltage signal to the motors, and $\boldsymbol{f}$ refers to the output force of the motors acting on the driving rods. The continuum robot shown in Figs. 4 and 7 will be a dynamic model in the simulation or a robotic prototype in the experiment. The motor can be considered as a proportional element since its bandwidth is much higher than that of the continuum robot. The PID controller is defined as

$$\mathbf{u} = k_p \cdot \delta\mathbf{l} + k_i \cdot \int (\delta\mathbf{l})dt + k_d \cdot \frac{d(\delta\mathbf{l})}{dt} \tag{9}$$

where k_p is the proportional coefficient, k_i is the integral coefficient and k_d is the derivative coefficient. The units for the PID coefficients are V/m, V/(m•s) and V/(m/s) respectively.

3.3.2. Comparison at different points with the same moving direction

In this group of simulations, two starting points with different position vectors (Point1:$[-0.0632, 0, 0.8665]^T$, Point2: $[-0.0406, 0, 0.8669]^T$) and configuration vectors (Point1:$[0, -\pi/64, 0, \pi/60]^T$, Point2: $[0, -\pi/64, 0, \pi/64]^T$) are selected for the robot tip to move by a small distance of 0.01 m in 0.1 s in the positive direction of the $\boldsymbol{X}$ axis.

The bottom right of Fig. 5 shows the positions of the two starting points in the workspace. Although these points are close, they have quite different velocity sensitivity,$|\boldsymbol{\Lambda_1}| = 16.7055$ and $|\boldsymbol{\Lambda_2}| = 72.5550$, in the $\boldsymbol{X}$ direction.

These PID coefficients in the servo controller are set to $k_p = 200$, $k_i = 25$ and $k_d = 4$ for both points in the simulations. Because the 2-norm of the velocity sensitivity $|\boldsymbol{\Lambda_2}|$ is 334.3% higher than $|\boldsymbol{\Lambda_1}|$, the simulated results show different effects. The movement starting from point 2 has obvious overshoot and oscillations, yet the movement starting from point 1 is stabler and slower. This indicates that the current PID coefficients are suitable for the control of point 1 where the velocity sensitivity is small, but too 'strong' for point 2 that has large velocity sensitivity. It is necessary to reduce the P gain and increase the D gain for point 2, to improve the damping and stability in the control process.

3.3.3. Comparison at the same point with different moving directions

In this group of simulations, a starting point in the workspace will be selected for the robot tip to move by a small distance of 0.01 m in 0.1 s along the $\boldsymbol{X}$ and $\boldsymbol{Y}$ directions, respectively. The position vector of the point A in the workspace is $[-0.1068, 0, 0.8496]^T$ while the configuration vector of the point in the configuration space is $[0, -\pi/24, 0, \pi/12]^T$. The 2-norm of the velocity sensitivity along the $\boldsymbol{X}$ and $\boldsymbol{Y}$ directions are $|\boldsymbol{\Lambda_x}| = 71.5481$ and $|\boldsymbol{\Lambda_y}| = 190.7764$, respectively. The coefficients in the servo controller are set to $k_p = 300$, $k_i = 40$ and $k_d = 10$ for both directions in this test. As shown in Fig. 6, because the 2-norm of the velocity sensitivity $|\boldsymbol{\Lambda_y}|$ is 166.6% higher than $|\boldsymbol{\Lambda_x}|$, the movement in the $\boldsymbol{X}$ direction is smoother than that in the $\boldsymbol{Y}$ direction. Again, this indicates that it is necessary to tune the PD coefficients to compensate for the variation of velocity sensitivities in different moving directions.

Through the above simulations it can be found that the control effects achieved by using the constant servo control parameters may have significant differences depending on the current velocity sensitivity. This is related not only to the configurations, but also the moving directions of the continuum robot. If the servo control parameters can be accordingly adjusted with respect to the change of the velocity sensitivity, the control performance of the robot can then be stabilized.

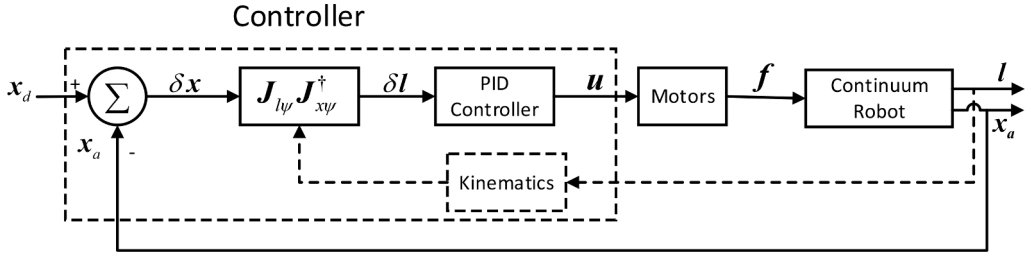


Fig. 4. Inverse kinematic control with PID Controller.

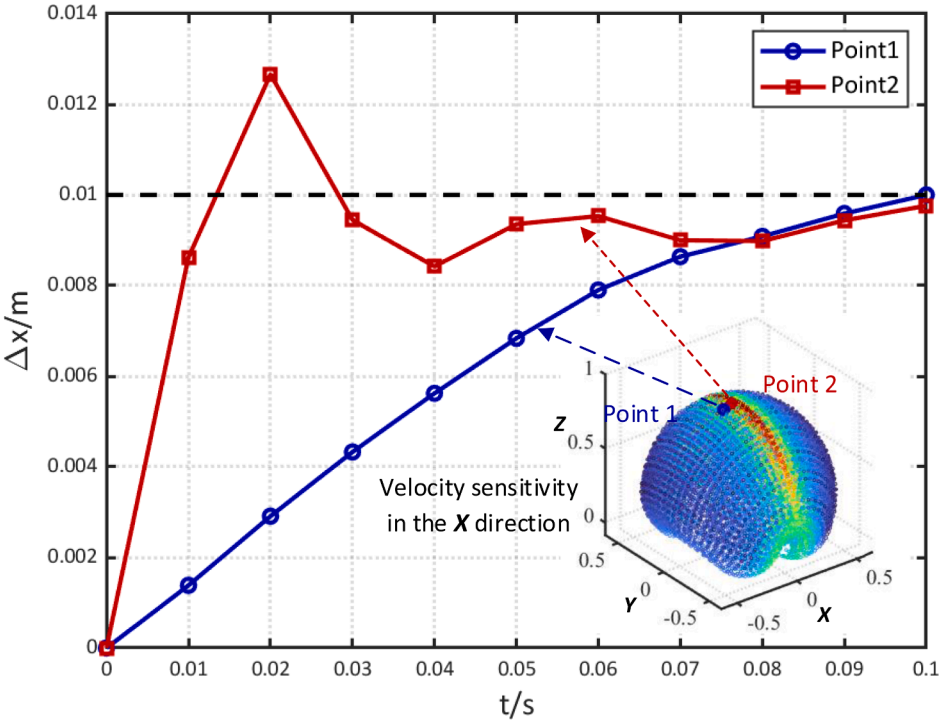


Fig. 5. The simulated movements of the end-tip starting at different points with the same direction.

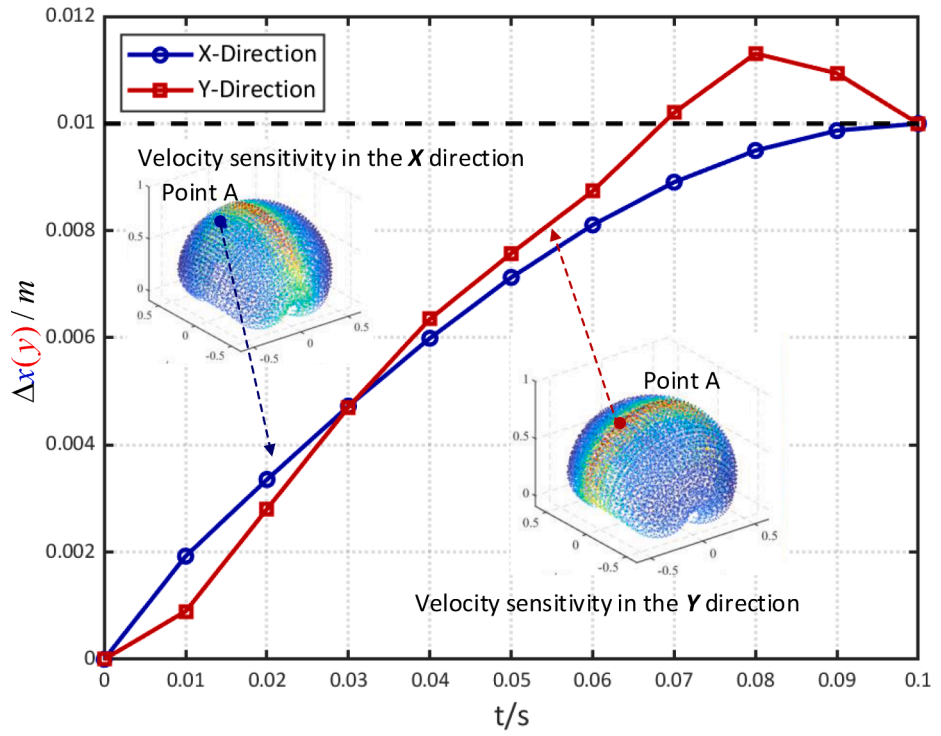


Fig. 6. The simulated movements of the end-tip starting at the same points with different directions.

4. Variable-gain control strategy

In this section, a control strategy based on the velocity sensitivity will be built to adjust the control parameters adaptively, and a series of simulations will be conducted to verify the effects of this strategy.

4.1. Design of the controller

Compared with the classic inverse kinematics control strategy presented in Fig. 4, a velocity sensitivity estimator is introduced to the control system, as shown in Fig. 7, and used to tune the PD coefficients, k_p and k_d, forming a variable gain controller (VG controller) that can be expressed by

$$\begin{cases} \mathbf{u} = k_p \cdot \delta\mathbf{l} + k_i \cdot \int (\delta\mathbf{l})dt + k_d \cdot \dfrac{d(\delta\mathbf{l})}{dt} \\ k_p = \dfrac{1}{|\boldsymbol{\Lambda}|} \cdot k_1 \\ k_d = |\boldsymbol{\Lambda}| \cdot k_2 \\ |\boldsymbol{\Lambda}| = \begin{cases} 0.1 & , \quad |\boldsymbol{\Lambda}| < 0.1 \\ |\boldsymbol{\Lambda}| & , \quad 0.1 \le |\boldsymbol{\Lambda}| \le 1000 \\ 1000 & , \quad |\boldsymbol{\Lambda}| > 1000 \end{cases} \end{cases} \tag{10}$$

where $\boldsymbol{\Lambda}$ is the velocity sensitivity calculated from the current configuration and the moving direction of the continuum robot, according to Eq. (8). The proportional coefficient k_p is reciprocal to the 2-norm of the velocity sensitivity, which means the proportional coefficient will be adaptively reduced if the robot has large velocity sensitivity. In the meantime, the derivative coefficient k_d is proportional to the 2-norm of the velocity sensitivity, so that the derivative effect (i.e. the damping effect) will be enhanced in the condition of high velocity sensitivity. The integral coefficient k_i is a constant here to reduce the steady state error, which is not the focus (i.e. suppressing oscillations) of this work. k_1 and k_2 are also constants to make the range of k_p and k_d within a reasonable level. They can be determined by simulated or experimental trials. At the same time, the necessary limits for $|\boldsymbol{\Lambda}|$ are defined to avoid instability in the controller.

The presented VG controller can provide a consistent control performance to the continuum robot, which means the robot will have less oscillations regardless of the changes in its configuration and moving direction. According to Eq. (10), the VG controller is actually on the basis of a PID control strategy, thus it's stability can be guaranteed through the PID algorithm and proved by using a Lyapunov function method [30].

4.2. Path tracking simulation

In this section, we will carry out two groups of path tracking simulations to verify the effects of the VG controller. In the first simulation, the desired path is selected to have large change in the velocity sensitivity along a straight line. In the second simulation, the desired path has a circular shape resulting in changing direction during the movement.

To evaluate the oscillation level in a path tracking process for the continuum robot, the fluctuation index $S(k)$ is introduced in this paper as

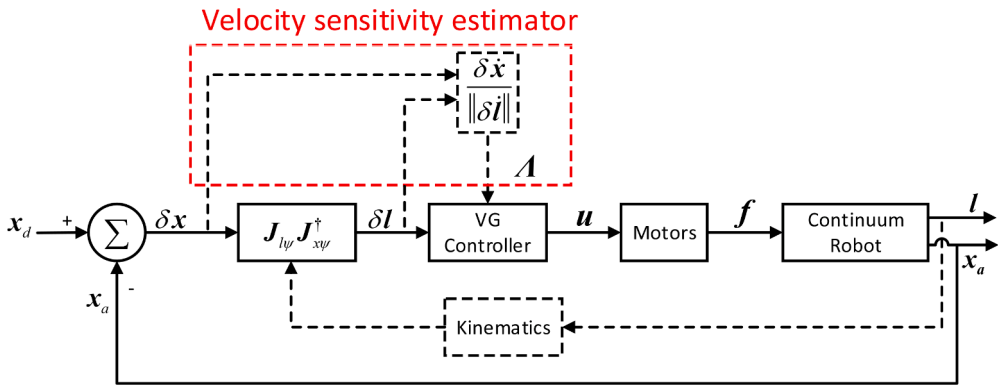


Fig. 7. Variable-gain control strategy.

$$S(k) = \left| \frac{\sum_{i=k-m}^{k} \left[\left[\left(\mathbf{x}_{a,i} - \mathbf{x}_{d,i} \right) - \boldsymbol{\mu} \right] \odot \left[\left(\mathbf{x}_{a,i} - \mathbf{x}_{d,i} \right) - \boldsymbol{\mu} \right] \right]}{m-1} \right| \tag{11}$$

where

$$\boldsymbol{\mu} = \frac{\sum_{i=k-m}^{k} \left(\mathbf{x}_{a,i} - \mathbf{x}_{d,i} \right)}{m} \tag{12}$$

Eqs. (11) and (12), $\mathbf{x}_{d,i}$ are the reference points in the desired path, $\mathbf{x}_{a,i}$ are the sampled points in the actual path, k indicates the time instant, m is the number of the sampled points before time instant k, and $\odot$ is the sign of Hadamard product [31]. Thus, the fluctuation index $S(k)$ defined here indicates the average level of the oscillations in a period between time instant k-m and k. Note that, a large value of m will lead to a low resolution of the fluctuation index S. In our work, $m = 5$ is selected to ensure a clear observation of the fluctuation index S. The control parameters used in the simulations are listed in Table 2, which can achieve the best control performance (i.e. the minimum S) in each case.

4.2.1. Tracking a straight line with changing configuration

To reflect the change in the velocity sensitivity due to the change in robot configuration, the desired path is selected as a straight line along the $\boldsymbol{X}$ direction on the $\boldsymbol{X}$–$\boldsymbol{Z}$ plane. The configuration vector of the starting point is $[0, -\pi/24, 0, -\pi/24]^T$ while that of the end point is $[0, \pi/24, 0, \pi/24]^T$. From the 2-norm of the velocity sensitivity plot shown in Fig. 8(a), it can be seen that the 2-norm of the velocity sensitivity on the desired path will firstly increase (moving from the blue area to the red area) and then decrease (moving from the red area to the blue area) along the $\boldsymbol{X}$ direction, which is consistent with the calculated results shown in Fig. 8(c). The actual tracking paths obtained by the PID and VG controllers are compared in Fig. 8(a). The VG controller shows good effect on reducing the oscillations.

As shown in Fig. 8(b), the use of the VG controller can decrease the fluctuation index S by 57.6% in comparison with the PID controller. Fig. 8(c) shows that the 2-norm of the velocity sensitivity along the desired path can change by 85.7%, from the starting point to the end point. Figs. 8(d) and (e) shows the corresponding changes in the control parameters k_p and k_d.

4.2.2. Tracking a circle with changing configuration and direction

In this simulation, the desired path is circle in the $\boldsymbol{X}$–$\boldsymbol{Y}$ plane, as shown in Fig. 9(a). The configuration vector of the starting point is $[-\pi/24, 0, \pi/8, 0]^T$, and the radius of the circle is 80 mm. Compared with the PID controller, although the maximum error of the VG controller is similar, the oscillations is greatly reduced. This is also proved by Fig. 9(b), in which the fluctuation index S decreases by 62.1% if using the VG controller. As shown in Fig. 9(c), (d) and (e), the control parameters k_p and k_d in the VG controller will change adaptively with respect to the change in the velocity sensitivity. This simulation shows again that the VG controller can significantly reduce the oscillations during a movement with changing configuration and direction.

5. Experimental validation

In this section, a prototype of the continuum robot was used to validated the presented VG controller by tracking a spiral path over the whole workspace.

5.1. Experimental platform

As shown in Fig. 10, the prototype of the continuum robot is composed of a 4- DOF continuum arm, an end effector and an actuation box. The continuum arm consists of two modules. Each module contains a backbone, three driving rods distributed 120 ° apart and 11 constraint disks. The total length of the arm is 870 mm, and the outer diameter is 38 mm. It is covered with a layer of nylon mesh which can limit the torsion deformation. A three-finger gripper is installed at the end of the arm as an end effector. An electromagnetic sensor (3D Guidance trakSTAR, Northern Digital Inc., Canada) is placed at the center of the gripper to feedback the end-tip position of the

Table 2
The control parameters used in the simulations.

Control strategy	Parameters	Tracking a straight line in Section 4.2.1	Tracking a circle in Section 4.2.2
PID controller	k_p	200	180
	k_i	23	19
	k_d	2.4	2.3
VG controller	k_1	13,000	10,500
	k_i	23	19
	k_2	0.035	0.040

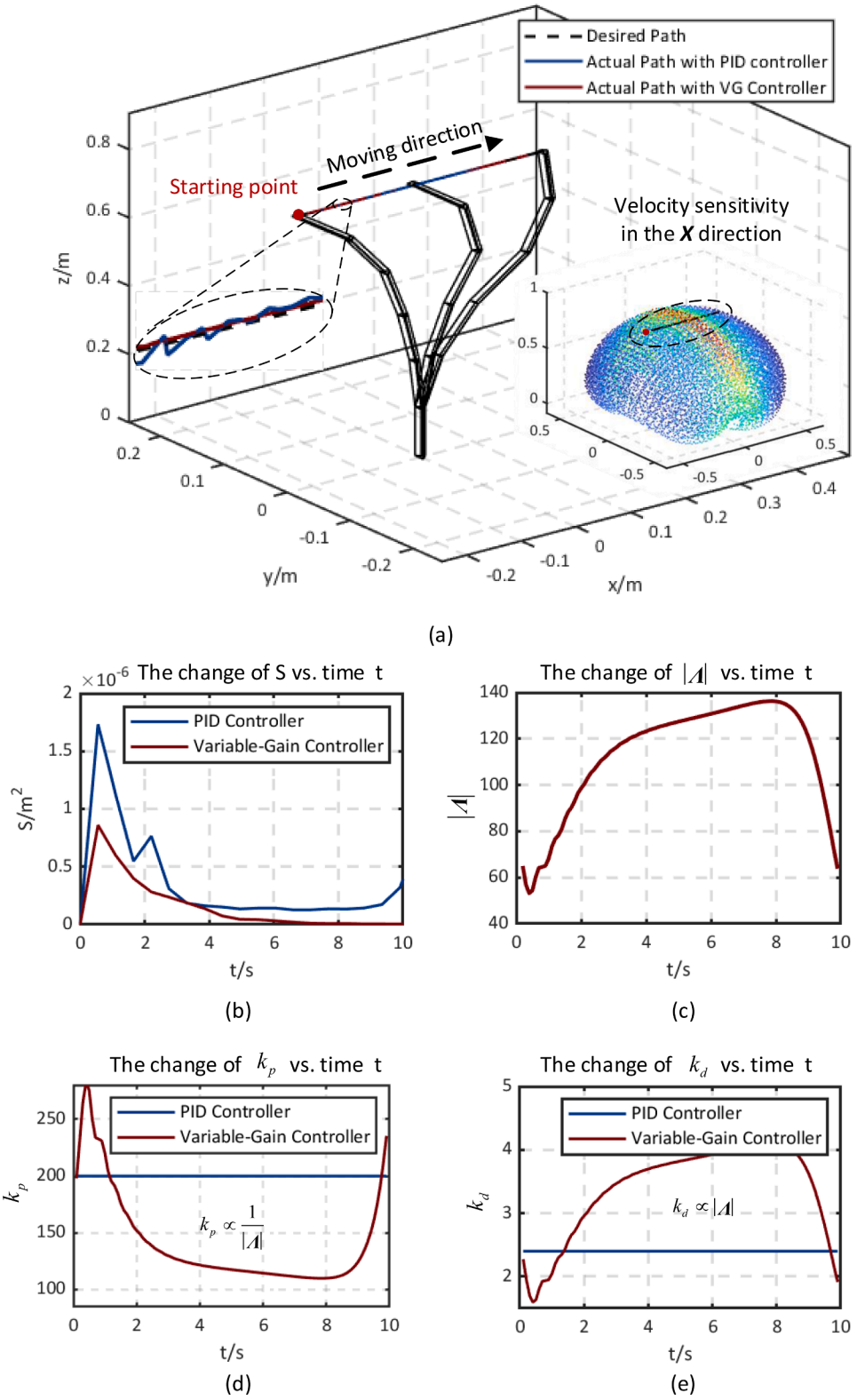


Fig. 8. Simulated results for tracking a straight line: (a)the tracking paths of using PID and VG controllers and the distribution of $|\Lambda|$ in the moving direction, (b) the change of fluctuation index S, (c) the change of the 2-norm of velocity sensitivity $|\Lambda|$, (f) the change of k_p, (e) the change of k_d.

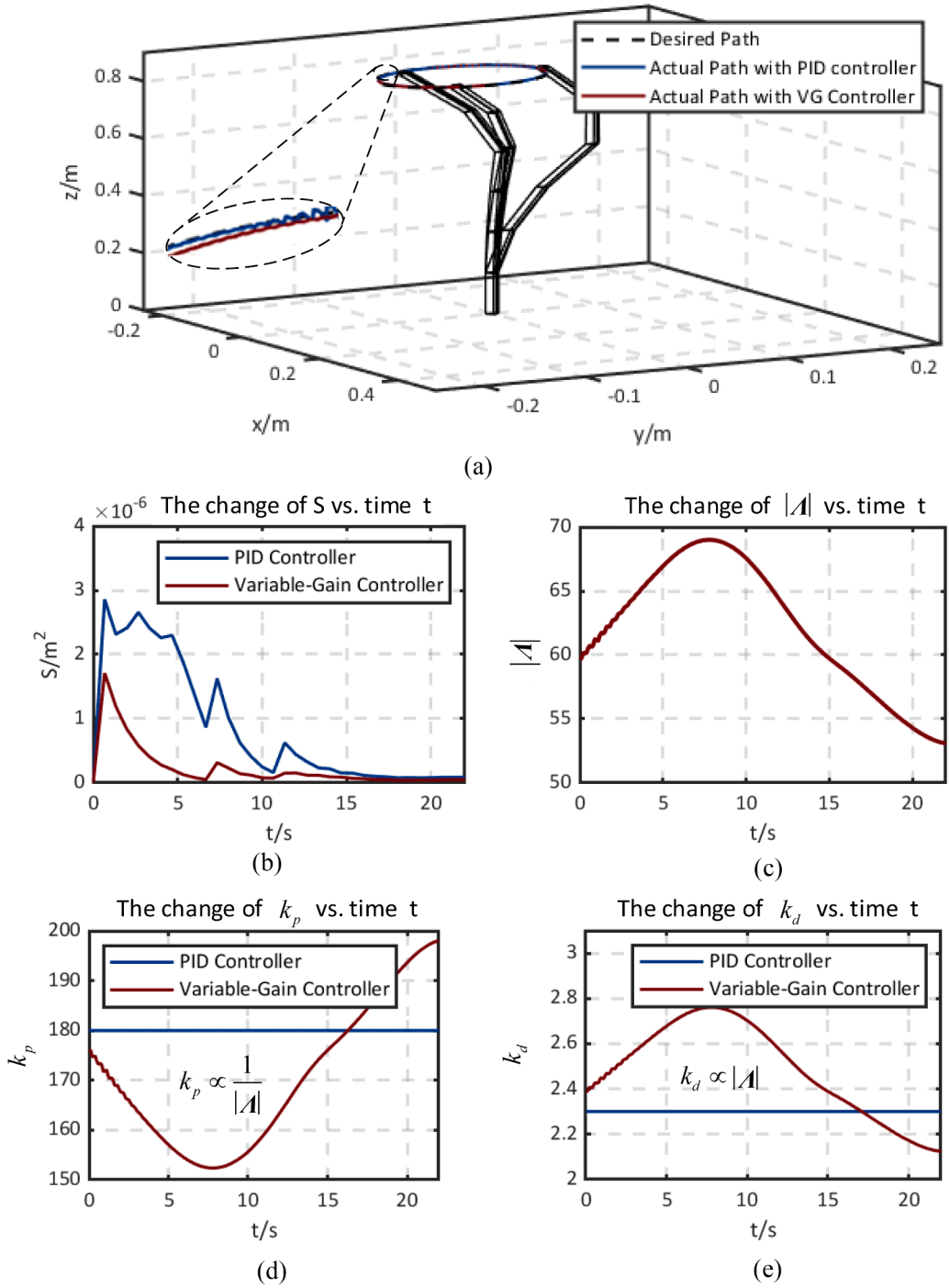


Fig. 9. Simulation results for tracking a circle: (a) a comparison of the tracking path, (b) the change of the fluctuation index S, (c) the change of the 2-norm of velocity sensitivity $|\Lambda|$, (d) the change of k_p, (e) the change of k_d.

arm. The actuation box includes six sets of DC motors and ball screws to pull or push the driving rod. Each DC motor is equipped with an optical encoder to measure the rotating angle for the use of calculating and feedbacking the actual length of each driving rod. Details of the robotic system can be found in [15].

5.2. Path tracking

In this section, the prototype is controlled to track a spiral path travelling across most areas of the workspace. Fig. 11(a) compares

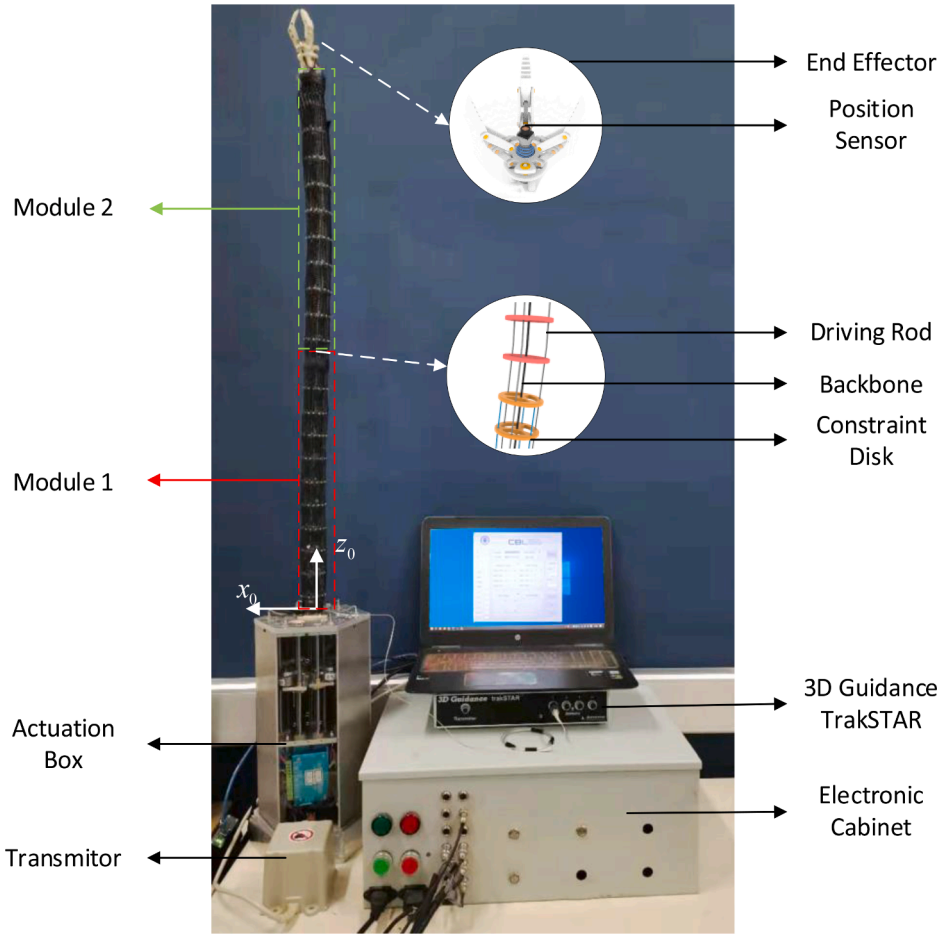


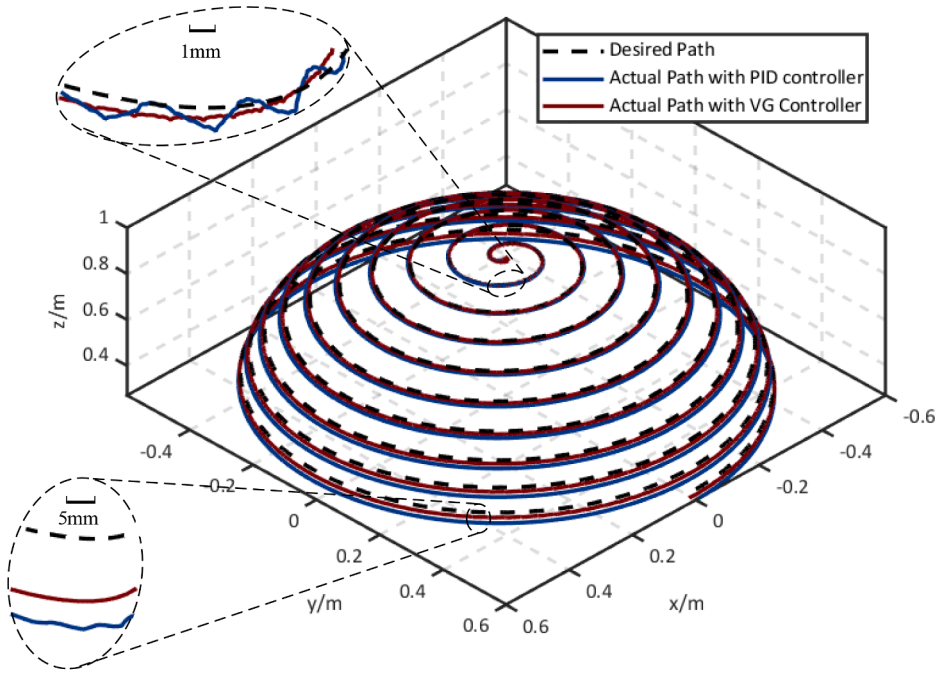
Fig. 10. Prototype of the continuum robot.

the actual paths obtained by the traditional PID controller and presented VG controller. It can be seen that the VG controller greatly reduced the oscillations during the movement. In either case of using PID or VG controller, the tracking error is increasing when the end-tip is reaching the bottom of the workspace. This is because the elastic deformation gets larger in those areas, resulting in strong stress to the robot body and making it difficult for the robot to achieve high accuracy. In this experiment, the proportional, integral and derivative coefficients k_p, k_i and k_d in the PID controller were set to 160, 20 and 1. The parameters k_1 and k_2 in the VG controller were set to 20,300 and 0.008, and the integral coefficient, k_i =20, in the VG controller is same as that in the traditional PID controller. With the help of VG controller, the fluctuation index S can be reduced by up to 76.4%, Fig. 11(b). In addition, it can be found that the fluctuation index has an increasing trend in the tracking process. This is because the velocity sensitivity is getting larger, Fig. 11(c). With the VG controller, the control parameters can be tuned adaptively, so the increasement of the fluctuation index is smaller than that of the traditional PID controller, Fig. 11(b). Fig. 11(d) and (e) shows the change of the proportional and derivative coefficients k_p and k_d in the VG controller according to the change of the 2-norm of the velocity sensitivity, Fig. 11(c).

6. Conclusions

In this paper, a kinematic parameter, velocity sensitivity, is proposed to evaluate the kinematic gain from the individual actuators to the end-tip position for a continuum robot. It was found that the velocity sensitivity will vary according to the change of the configuration and moving direction of the robot. Based on this, a new variable gain control strategy at the servo control level was developed to compensate for the variation of velocity sensitivity at the kinematic control level. This variable gain control strategy allows to adjust the servo control parameters, e.g. the proportional and derivative coefficients in our case, adaptively during a tracking process.

The effects of the variable-gain control strategy on the continuum robot was verified by simulations and experiments that include some typical working conditions for the robot. The fluctuation index is presented to evaluate the oscillations in the tracking path. The experimental results show that the presented variable gain control strategy can significantly reduce the fluctuation index by 76.4% in comparison with the traditional PID controller in most areas in the workspace.



(a)

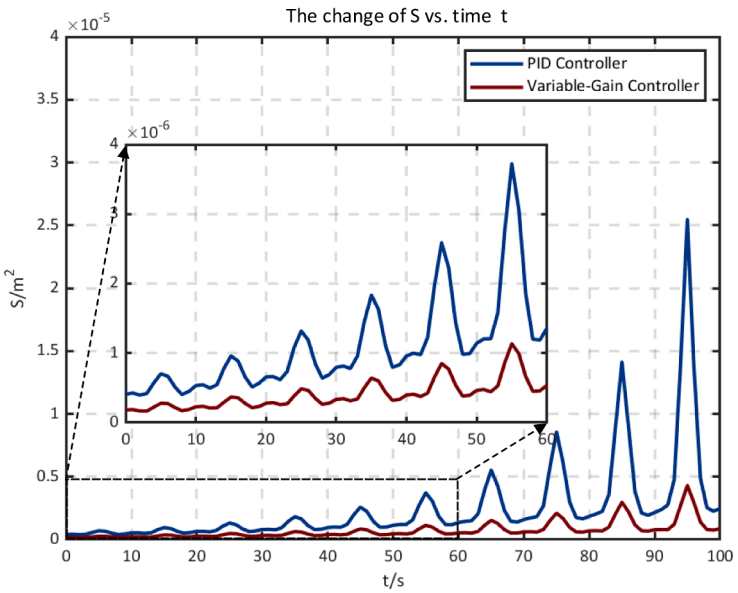


Fig. 11. Experimental results: (a) a comparison between the PID and VG controllers, (b) the change of the fluctuation index S, (c) the change of the 2-norm of the velocity sensitivity $|A|$, (d) the change of k_p, (e) the change of k_d.

Funding

This work was supported by the Natural Science Foundation of China (Grant No. 51875393), National Key R&D Program of China (Grant No. 2018YFB1304600 and 2019YFB1309800), the Tianjin Key Program of Science and Technology (Grant No. 19YFZCCG00410), and the State Key Laboratory of Robotics Foundation-China (Grant No. 2019-O04).

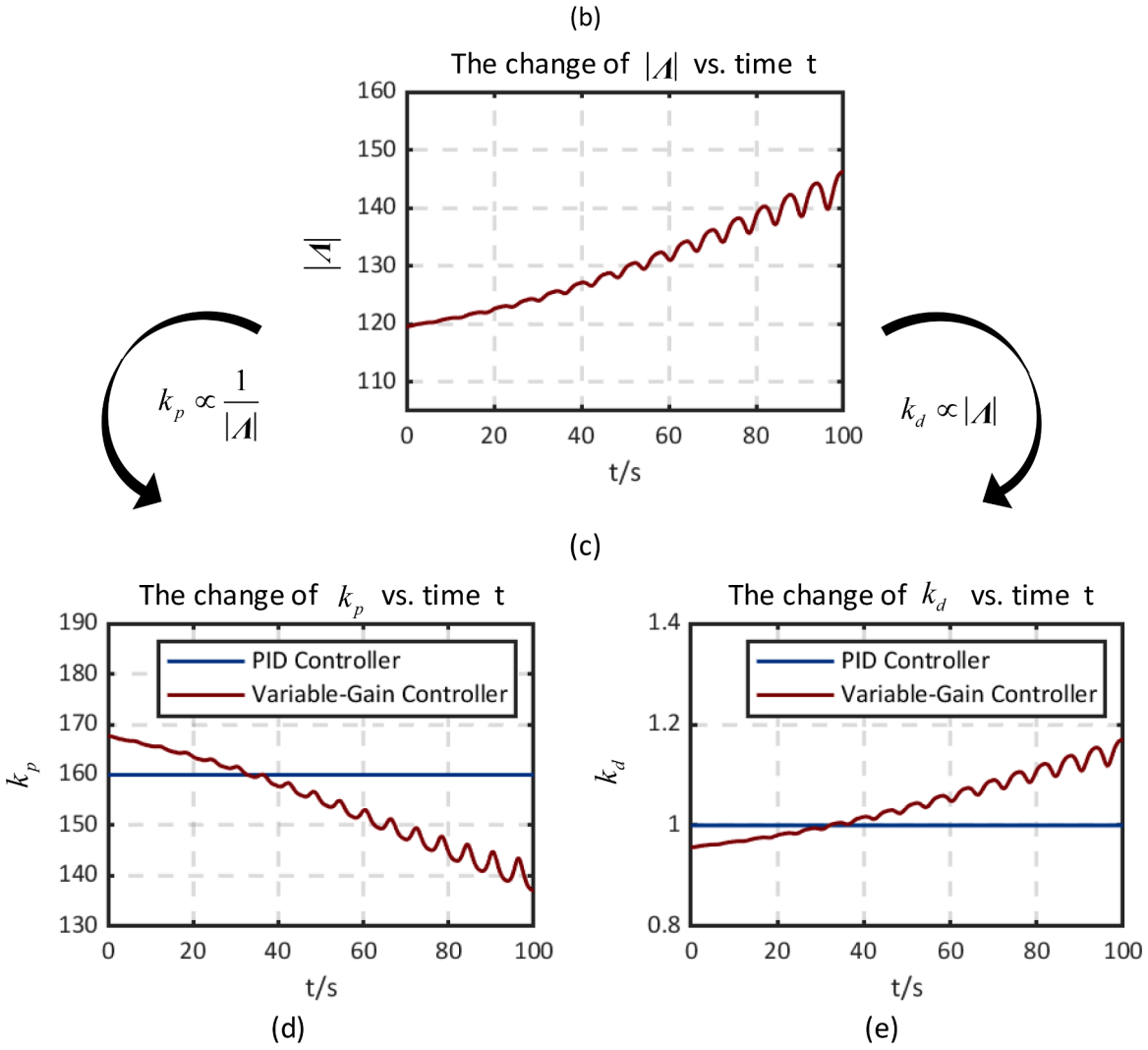


Fig. 11. (*continued*).

Declaration of Competing Interest

None

Supplementary materials

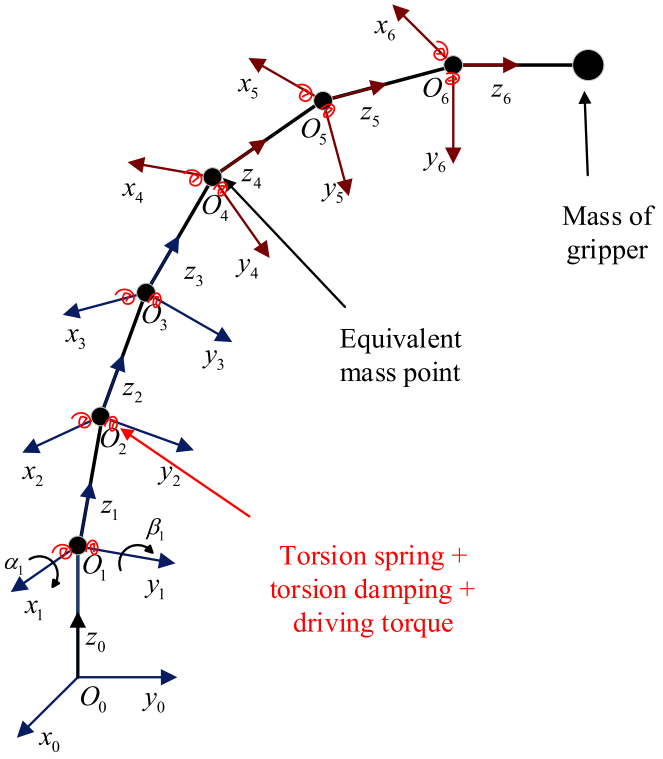
Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.mechmachtheory.2021.104618.

Appendix. : Dynamic of continuum robots derivation process

The mass of each equivalent mass point is defined as $m_i (i = 1,2,\ldots,6)$, the mass of a gripper m_t. The distance between the fixed base and the moving platform is known. The distance between the centroid of gripper and the top moving platform is known too. The Euler angles of each parallel platform are α_j and $\beta_j (j = 1,2)$. The elastic coefficients of the torsion spring in the dynamic model are k_x and k_y. The damping coefficients of torsional damping are c_x and c_y. The driving torque is T_x^i and T_y^i.

- Kinetic energy

The coordinates of each particle can be obtained from the kinematic model of the continuum robot $\mathbf{x}_i$ and $\mathbf{x}_t$. So the kinetic energy



Extended Data Fig. S1. . Dynamic of continuum robots.

can be obtained as

$$K = \sum_{i=1}^{6} \frac{1}{2} m_i \dot{\mathbf{x}}_i^{\,2} + \frac{1}{2} m_t \dot{\mathbf{x}}_t^{\,2} \tag{13}$$

- Gravitational potential energy

The $\boldsymbol{Z}$ coordinates of each particle can be obtained from the kinematic model of the continuum robot z_i and z_t. So the gravitational potential energy can be obtained as (The gravitational potential energy at O_0 is defined as 0)

$$P_z = \sum_{i=1}^{6} m_i g z_i + m_t g z_t \tag{14}$$

- Elastic potential energy

The elastic potential energy of the system is the energy of each torsion spring. So the elastic potential energy can be obtained as

$$P_t = \sum_{i=1}^{6} \left(\frac{1}{2} k_x \alpha_i^{\,2} + \frac{1}{2} k_y \beta_i^{\,2} \right) \tag{15}$$

Then the Lagrange equation is

$$L = K - P_z - P_t$$

So T_x^i and T_y^i. can be obtained

$$\left.\begin{aligned} T_x^i - c_x\dot{\alpha}_i &= \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\alpha}_i}\right) - \frac{\partial L}{\partial \alpha_i} \\ T_y^i - c_y\dot{\beta}_i &= \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\beta}_i}\right) - \frac{\partial L}{\partial \beta_i} \end{aligned}\right\} \tag{16}$$

It can be written as Eq. (7). In the Eq. (7)

$$\mathbf{T} = \begin{bmatrix} T_x^1 & T_y^1 & T_x^2 & T_y^2 \end{bmatrix}^T \tag{17}$$

$$\mathbf{M} = \begin{bmatrix} \frac{\partial}{\partial \dot{\alpha}_1}\left(\frac{\partial L}{\partial \dot{\alpha}_1}\right) & \frac{\partial}{\partial \dot{\beta}_1}\left(\frac{\partial L}{\partial \dot{\alpha}_1}\right) & \frac{\partial}{\partial \dot{\alpha}_2}\left(\frac{\partial L}{\partial \dot{\alpha}_1}\right) & \frac{\partial}{\partial \dot{\beta}_2}\left(\frac{\partial L}{\partial \dot{\alpha}_1}\right) \\ \frac{\partial}{\partial \dot{\alpha}_1}\left(\frac{\partial L}{\partial \dot{\beta}_1}\right) & \frac{\partial}{\partial \dot{\beta}_1}\left(\frac{\partial L}{\partial \dot{\beta}_1}\right) & \frac{\partial}{\partial \dot{\alpha}_2}\left(\frac{\partial L}{\partial \dot{\beta}_1}\right) & \frac{\partial}{\partial \dot{\beta}_2}\left(\frac{\partial L}{\partial \dot{\beta}_1}\right) \\ \frac{\partial}{\partial \dot{\alpha}_1}\left(\frac{\partial L}{\partial \dot{\alpha}_2}\right) & \frac{\partial}{\partial \dot{\beta}_1}\left(\frac{\partial L}{\partial \dot{\alpha}_2}\right) & \frac{\partial}{\partial \dot{\alpha}_2}\left(\frac{\partial L}{\partial \dot{\alpha}_2}\right) & \frac{\partial}{\partial \dot{\beta}_2}\left(\frac{\partial L}{\partial \dot{\alpha}_2}\right) \\ \frac{\partial}{\partial \dot{\alpha}_1}\left(\frac{\partial L}{\partial \dot{\beta}_2}\right) & \frac{\partial}{\partial \dot{\beta}_1}\left(\frac{\partial L}{\partial \dot{\beta}_2}\right) & \frac{\partial}{\partial \dot{\alpha}_2}\left(\frac{\partial L}{\partial \dot{\beta}_2}\right) & \frac{\partial}{\partial \dot{\beta}_2}\left(\frac{\partial L}{\partial \dot{\beta}_2}\right) \end{bmatrix} \tag{18}$$

$$\mathbf{C} = \begin{bmatrix} \frac{\partial}{\partial \alpha_1}\left(\frac{\partial L}{\partial \dot{\alpha}_1}\right) + c_x & \frac{\partial}{\partial \beta_1}\left(\frac{\partial L}{\partial \dot{\alpha}_1}\right) & \frac{\partial}{\partial \alpha_2}\left(\frac{\partial L}{\partial \dot{\alpha}_1}\right) & \frac{\partial}{\partial \beta_2}\left(\frac{\partial L}{\partial \dot{\alpha}_1}\right) \\ \frac{\partial}{\partial \alpha_1}\left(\frac{\partial L}{\partial \dot{\beta}_1}\right) & \frac{\partial}{\partial \beta_1}\left(\frac{\partial L}{\partial \dot{\beta}_1}\right) + c_y & \frac{\partial}{\partial \alpha_2}\left(\frac{\partial L}{\partial \dot{\beta}_1}\right) & \frac{\partial}{\partial \beta_2}\left(\frac{\partial L}{\partial \dot{\beta}_1}\right) \\ \frac{\partial}{\partial \alpha_1}\left(\frac{\partial L}{\partial \dot{\alpha}_2}\right) & \frac{\partial}{\partial \beta_1}\left(\frac{\partial L}{\partial \dot{\alpha}_2}\right) & \frac{\partial}{\partial \alpha_2}\left(\frac{\partial L}{\partial \dot{\alpha}_2}\right) + c_x & \frac{\partial}{\partial \beta_2}\left(\frac{\partial L}{\partial \dot{\alpha}_2}\right) \\ \frac{\partial}{\partial \alpha_1}\left(\frac{\partial L}{\partial \dot{\beta}_2}\right) & \frac{\partial}{\partial \beta_1}\left(\frac{\partial L}{\partial \dot{\beta}_2}\right) & \frac{\partial}{\partial \alpha_2}\left(\frac{\partial L}{\partial \dot{\beta}_2}\right) & \frac{\partial}{\partial \beta_2}\left(\frac{\partial L}{\partial \dot{\beta}_2}\right) + c_y \end{bmatrix} \tag{19}$$

$$\mathbf{X} = \begin{bmatrix} \frac{\partial L}{\partial \alpha_1} & \frac{\partial L}{\partial \beta_1} & \frac{\partial L}{\partial \alpha_2} & \frac{\partial L}{\partial \beta_2} \end{bmatrix}^T \tag{20}$$

References

[1] J.J. Craig, Introduction to Robotics: Mechanics and Control, 3/E, Pearson Education India, 2009.
[2] A.D. Souza, S. Vijayakumar, S. Schaal, Learning inverse kinematics, in: IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium, 2001, pp. 298–303.
[3] G. Tevatia, S. Schaal, Inverse kinematics for humanoid robots, in: IEEE International Conference on Robotics and Automation, 2000, pp. 294–299.
[4] P. Rocco, Stability of PID control for industrial robot arms, IEEE T Robot. Autom. 12 (1996) 606–614.
[5] P. Berthet-Rayne, K. Leibrandt, G. Gras, P. Fraisse, A. Crosnier, G. Yang, Inverse kinematics control methods for redundant snakelike robot teleoperation during minimally invasive surgery, IEEE Robot. Autom. Lett. 3 (2018) 2501–2508.
[6] Z. Mao, T.C. Hsia, Obstacle avoidance inverse kinematics solution of redundant robots by neural networks, Robotica 15 (1997) 3–10.
[7] M. Li, R. Kang, S. Geng, E. Guglielmino, Design and control of a tendon-driven continuum robot, T I Meas. Control 40 (2017) 3263–3272.
[8] C. Ma, T. Zhao, G. Xiang, J. Ren, Y. Chen, S. Dian, Gaussian process based differential kinematics control of the soft robot, J. Mech. Eng. 57 (2021) 163–171.
[9] J.K. Parker, A.R. Khoogar, D.E. Goldberg, Inverse kinematics of redundant robots using genetic algorithms, in: IEEE International Conference on Robotics and Automation, IEEE Computer Society, 1989, pp. 271–276.
[10] N. Adhikary, C. Mahanta, Sliding mode control of position commanded robot manipulators, Control Eng. Pract. 81 (2018) 183–198.
[11] S.G. Anavatti, S.A. Salman, J.y. Choi, Fuzzy + PID controller for robot manipulator, in: International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce, 2006, p. 75. -75.
[12] G. Iacca, F. Caraffini, F. Neri, Memory-saving memetic computing for path-following mobile robots, Appl. Soft Comput. 13 (2013) 2003–2016.

[13] D.P. Kwok, S. Fang, Genetic algorithm and simulated annealing for optimal robot arm PID control, in: Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, 1994, pp. 707–713.
[14] Q. Li, Z. Cheng, J. Qian, Self-learning fuzzy PID controller based on neural networks, in: American Control Conference, IEEE, 1998, pp. 1860–1861.
[15] C. Yang, S. Geng, I. Walker, D.T. Branson, J. Liu, J.S. Dai, R. Kang, Geometric constraint-based modeling and analysis of a novel continuum robot with shape memory alloy initiated variable stiffness, Int J Rob Res 39 (2020) 1620–1634.
[16] J.K. Salisbury, J.J. Craig, Articulated Hands: force Control and Kinematic Issues, Int. J. Rob. Res. 1 (1982) 4–17.
[17] T. Yoshikawa, Analysis and control of redundant manipulators with redundancy, in: Robot Research, The First International Symposium, 1983.
[18] T. Yoshikawa, Manipulability of robotic mechanisms, Int. J. Rob. Res. 4 (1985) 3–9.
[19] Q. Cao, M. Sun, W. Xue, S. Xia, Research on avoidance of dynamic obstacle and singularity for manipulator in human-robot cooperation, J. Huazhong Univ. Sci. Technol. (Nat. Sci. Ed.) 48 (2020) 55–59. +65.
[20] H.-.G. Kim, K.-.S. Shin, S.-.W. Hwang, C.-.S. Han, Link length determination method for the reduction of the performance deviation of the manipulator: extension of the valid workspace, Int. J. Precis Eng. Man. 15 (2014) 1831–1838.
[21] H. Zhang, S. Liu, Y. Jiang, T. Wang, F. Jing, Research on mobile manipulator path planning based on operability, Autom. Instrum. 33 (2018) 45–49.
[22] Q. Chen, Y. Wang, H. Chen, Comparative research of trajectory tracking performance of robotic manipulator based on PD control scheme, Control Decis. 1 (2003) 53–57.
[23] J.A. Saglia, J.S. Dai, Geometry and kinematic analysis of a redundantly actuated parallel mechanism for rehabilitation, 2007, pp. 1081–1090.
[24] Automatic supervisory control of the configuration and behavior of multibody mechanisms, IEEE Trans. Syst. Man Cybern. 7 (1977) 868–871.
[25] R. Kang, D.T. Branson, T. Zheng, E. Guglielmino, D.G. Caldwell, Design, modeling and control of a pneumatically actuated manipulator inspired by biological continuum structures, Bioinspir. Biomim. 8 (2013) 8–36.
[26] R. Kang, Y. Guo, L. Chen, D.T. Branson, J.S. Dai, Design of a pneumatic muscle based continuum robot with embedded tendons, IEEE/ASME Trans. Mechatron. 22 (2017) 751–761.
[27] M. Li, R. Kang, D.T. Branson, J.S. Dai, Model-Free control for continuum robots based on an adaptive kalman filter, IEEE/ASME Trans. Mechatron. 23 (2018) 286–297.
[28] W.S. Rone, P. Ben-Tzvi, Continuum robot dynamics utilizing the principle of virtual power, IEEE T. Robot. 30 (2014) 275–287.
[29] R. Kang, D.T. Branson, E. Guglielmino, D.G. Caldwell, Dynamic modeling and control of an octopus inspired multiple continuum arm robot, Comput. Math. Appl. 64 (2012) 1004–1016.
[30] B. Armstrong, J. McPherson, Y. Li, On the stability of nonlinear PD control, Appl. Math. Comput. Sci. 7 (1997) 273–292.
[31] T. Ando, Majorization relations for Hadamard products, Linear Algebra Appl. 223 (1995) 57–64.