PAPER

Adaptive neural controller with predefined boundary constraints for 3D printed cable-driven soft actuators

IOP Publishing Smart Materials and Structures
Smart Mater. Struct. **32** (2023) 065012 (12pp) https://doi.org/10.1088/1361-665X/acd157

Adaptive neural controller with predefined boundary constraints for 3D printed cable-driven soft actuators

Chengkai Xia[1,2], Zhenbang Xu[2,*], Huayang Sai[1,2], Hang Li[2] and Kai Wang[1,2]

[1] CAS Key Laboratory of On-orbit Manufacturing and Integration for Space Optics System, Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, People's Republic of China
[2] University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China

E-mail: xuzhenbang@ciomp.ac.cn

Received 14 December 2022, revised 25 April 2023
Accepted for publication 28 April 2023
Published 9 May 2023

Abstract

It is difficult to build dynamic models for 3D-printed soft actuators because of their material and structural flexibility and the complex intrinsic and extrinsic interactions encountered in human-centric or complex non-structural environments. Soft actuators require the control error during motion to be limited. However, existing control methods with predefined boundary constraints are designed for rigid actuators and are not directly applicable to soft actuators. In this paper, an adaptive neural controller based on a quasi-static model is proposed. The quasi-static model of the soft actuator is used to determine how the viscoelasticity of the flexible material influences the neural network, enabling the neural network to identify a better fit to the dynamic model of the soft actuator. Finally, experimental results verify that the proposed controller constrains the tracking error of the soft actuator to within the predefined boundary.

Keywords: soft actuator, adaptive neural controller, predefined boundary constraints, Euler–Bernoulli beam

(Some figures may appear in colour only in the online journal)

1. Introduction

Inspired by natural biology, soft robotics aims to reproduce biological softness and a soft structure with infinite degrees of freedom (DoF), resulting in more adaptable, robust, and effective robots. Soft actuators have intrinsic flexural compliance, which avoids the need for fine kinematic control during contact with fragile objects [1, 2]. They combine compliant machinery and flexible drivers to provide better solutions than traditional rigid industrial robots in natural and human-centric environments, where safety and adaptability to uncertainty are fundamental [3].

In recent decades, many soft robotic arms have been developed [4–9]. In science and processing, especially 3D printing, several smart materials and structures have achieved greater levels of machinability, such as dielectric elastomers [10, 11], shape memory alloys [12], and other soft actuators [13]. However, the viscoelasticity of flexible materials makes existing control methods unsuitable. Some researchers have used fluidic actuators in their soft robotic arms to achieve bending motions [4, 5, 9, 14–16], but the fluid is difficult to control precisely and fine manipulation remains challenging. Another straightforward method for controlling soft robots is cable-driven actuation, whereby the bending motion of soft robotic arms can be precisely controlled by pulling a cable [17–20]. The advantage of this approach is that the proven technology of traditional rigid robots can be used to precisely

* Author to whom any correspondence should be addressed.

control the length of the cable, allowing accurate bending control of the soft actuator.

The flexibility and viscoelasticity of materials and the nonlinearity of systems involving complex structures pose several new challenges [21]. Current research can be divided into two main strands: model-based and non-model-based. Efforts have been devoted to developing a valid and reliable model for flexible actuators such as lumped-mass models, finite element methods [22, 23], Cosserat rod models [24], kinematics-based approaches such as constant-curvature models, Denavit–Hartenberg parameter models [25, 26] and serial-link Lagrangian models. This problem has been considered in the study of continuum robotic arms approximating to soft arms with the aim of improving the modeling accuracy. The static and dynamic control of a continuum robot with a single cross-section has been analyzed [27], where three-dimensional deformation was achieved and modeled by passing a cable through a general curved path. A static model of a continuum robot that considers most of the common forces and moments applied by its own actuators and the external environment has also been developed [28]. However, these models can only handle actuators with simple structures and do not consider the properties of flexible materials. Non-model-based methods mainly rely on machine learning to solve nonlinear problems [29]. These approaches circumvent the need to define the parameters of the configuration space and are independent of the specific structure of the actuator. The complex kinematic or dynamical model is completely fitted by a training process that requires sufficient sample data. This approach usually achieves better performance in unstructured environments that are highly nonlinear, nonuniform, or difficult to model [30]. Particularly in the case of underwater robots, model-based methods may involve complex fluid dynamics modeling and analysis [31, 32]. Furthermore, the black box nature of machine learning makes stability analysis and convergence proofs difficult to establish [21].

Conventional control methods based on actuator dynamic models can effectively exploit the performance of the actuators [33]. However, the complex structures of 3D printing mean that accurate dynamic models are challenging. This makes it difficult to achieve trajectory tracking and output force control using traditional model-based control algorithms. Adaptive neural controllers have been developed to deal with such complex scenarios [34, 35]. To guarantee that the tracking error ultimately converges to within the predetermined boundaries, Bechlioulis and Rovithakis [36, 37] proposed a significant error transformation method, called 'prescribed performance', for the single-input/single-output strict-feedback nonlinear systems and the multiple-input/multiple-output affine nonlinear systems. Prescribed performance refers to the output error converging to a predefined arbitrarily small residual set, with convergence rate no less than a certain prespecified value, and a maximum overshoot of less than a preassigned level. Radial basis function neural networks (RBFNNs) are capable of approximating arbitrary continuous functions with arbitrary accuracy. Their application to the control of soft actuators can be used to fit complex nonlinear models. The network can be combined with an adaptive controller, and can solve the problem of non-strict-feedback nonlinear systems with input delays and prescribed boundary constraints [38, 39]. This makes it possible to achieve high-precision tracking of flexible actuator trajectories. However, the viscous differentiation of 3D-printed flexible resin compared with common damping in industrial materials creates a significant delay in the position feedback on the controller side. Adaptive controllers may generate large tracking errors or even non-convergence errors when fitting such large delays, leading to tracking failure. Therefore, developing an adaptive controller that is not based on a model of soft actuator dynamics remains a great challenge.

In this paper, an experimentally validated neural adaptive controller based on Euler–Bernoulli beam theory with prescribed boundary constraints is proposed. The controller is shown to achieve high-precision trajectory tracking of cable-driven soft actuators. The main contributions of this study are as follows.

- The designed control method contains an adaptive neural network and a quasi-static model of the soft actuator for fitting the dynamics of the system in operation, so that the controller is not based on the dynamics of the system. This avoids the need to model the complex dynamics of the soft robotic arm.
- An experimentally validated Euler–Bernoulli quasi-static model is used to calculate the stable position in the current drive state. Thus eliminating the output instability caused by unstable position delay and addressing the issue of the limited applicability of predefined boundary constraint controllers in soft actuator systems.
- Through a target trajectory tracking experiment, we show that the position of the end of the flexible actuator can quickly follow the set trajectory during the motion. This proves that the proposed control method achieves accurate trajectory tracking of a cable-driven flexible actuator.

Section 2 introduces the design of the soft actuator structure used for testing. Section 3 describes the proposed and validated quasi-static model used for the control. Section 4 presents the designed controller and section 5 verifies that this controller effectively implements the trajectory tracking of the soft actuator through a series of experiments. Finally, section 6 concludes with comments on the potential applications of the proposed controller and some ideas for future work.

2. Soft actuator design

The design of the soft actuator is based on several criteria: flexibility, reliability, and ease of manufacturing. In this study, we use silicone resin, a 3D-printable and elastomeric material. The soft actuator consists of two parts, the soft body and the drive system.

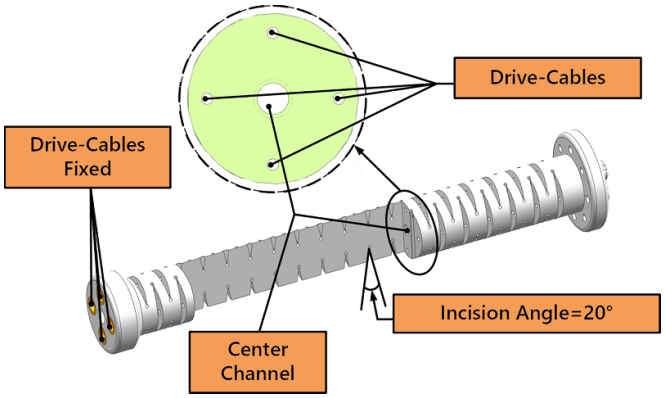

Figure 1. Construction of the soft actuator.

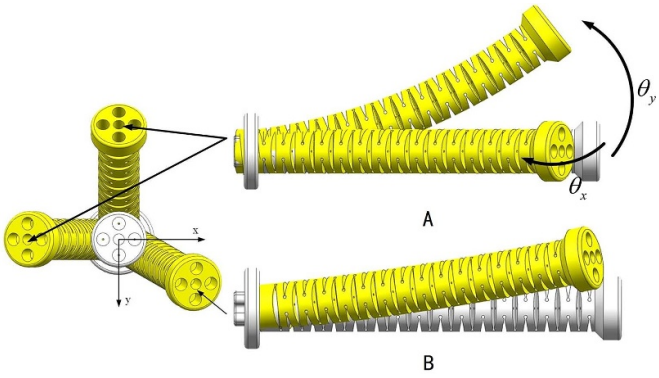

Figure 2. Bending diagram of the soft arm. (A) Two states of driving with single DoF. (B) Driving using two DoFs simultaneously.

2.1. Soft body design

Given the criteria stated above, a cable-driven mechanism is selected for our design. In view of flexibility, the 3D-printed body is designed to be coupled with lateral dentation structures on four sides, the shape of which was optimized via finite element simulation. Following optimization, the actuator could achieve a bending angle of approximately 180° when subjected to a 30 N drive force, with the maximum equivalent stress on each groove constituting around 80% of the tensile strength of FLAX9795. This design feature fully capitalizes on the flexibility of the material. The primary structure of the soft actuator was fabricated using FLAX9795, a photosensitive silicone resin with mechanical properties akin to those of silicone rubber. Specifically, FLAX9795 exhibits a Young's modulus of approximately 10 MPa, a tensile strength of 3 MPa, and strong viscoelastic behavior. Each dentation structure has a hole that can be threaded with the cable. The 3D-printed body is actuated by power transmission elements. As discussed in the Introduction, the cable-driven mechanism is selected for our design. However, to achieve multi-directional bending in conventional rigid-link cable-driven robots or actuators, several cables must pass through the lumen. One of the most striking features of a soft actuator is that it cannot maintain its shape as a rigid-link actuator. Therefore, the shape and length of the driven cables in the actuator may not reflect its state, which complicates the control system. In contrast, our design allows cables to pass through via-holes, which constrain these cables to the approximate shape of the 3D-printed body. These four cables with central symmetry provide two DoFs. The cables are made of Dyneema, and each can bear more than 400 N. As Dyneema is not as sharp as metal cable, there is less danger of cutting the 3D-printed body. Figure 1 shows the 3D model of the soft actuator.

During the movement of the robotic actuator, two symmetrical drive lines move simultaneously. This generates one DoF in each direction, and the soft actuators can be bent over 180° ($\theta_x, \theta_y \geqslant 180°$), as shown in figure 2(A). When four drive lines are moved simultaneously, the actuator will move in a specified direction, as shown in figure 2(B). Therefore, there are two bending DoFs during the movement of the system, and they can generate a certain deflection angle to expand the actuator's working space. The soft actuator has complete environmental compliance. Additionally, we have eliminated the rigid actuator's multi-component assembly, making the movement process more coherent and compact. According to the traditional definition, this soft actuator is continuously bendable to contain an infinite number of DoFs. In this paper, we only consider the actuator to have two bending DoFs in a broad sense. To be precise, this soft actuator is a complete under-actuated system.

2.2. Drive system design

As shown in figure 3, each drive unit is designed to drag a cable independently. The drive system consists of 12 drive units, a transmission mechanism, and motors. It is placed internally to ensure a compact and efficient robot. Since the soft actuators are designed to be used in series, this drive system allows for the scalability of up to twelve drive units, with driving a maximum of three soft actuators, thus providing up to six controllable DoFs. In figure 3(D), each unit consists of a coupling, bearing seat, ball screw, linear slide, cable fixture, connection nut, and supporting board. The control system includes a microcontroller and the drivers required to operate the motor, as well as the power supply. The microprocessor controls the motors, and the motors drive the ball screws to drag the Dyneema lines. The bearing seat and linear slide are mounted on the supporting board, and the ball screw is set on the bearing seat with couplings attached. The 3D-printed body is mounted on strut members and be fixed with four screws. Each driven cable is attached to a connector through the center bore, with a chamfer to prevent them from breaking. The connector is fixed with a corresponding ball screw nut. As shown in figure 3(B), all transmission units are assembled on the pedestal and evenly distributed along the circumference. When the actuation system works, the cables remain stretched and tight through the control of the motors. As a result, the motor movements change the cable lengths, causing the soft actuator to move.

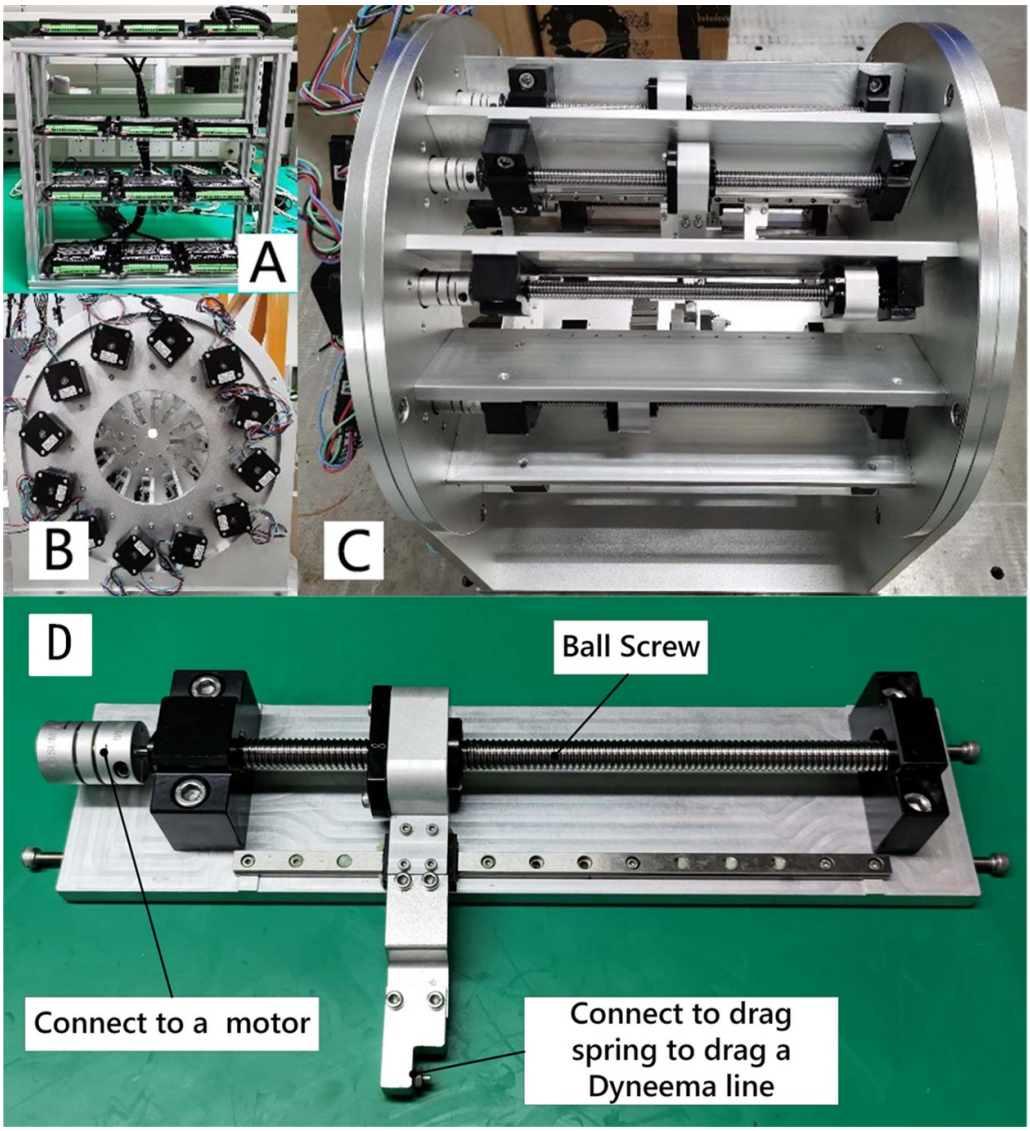

Figure 3. Drive system. (A) Drives of motors. (B) Twelve motors mounted on the supporting board. (C) Twelve drive units mounted on the supporting board. (D) Drive unit.

3. Theoretical modeling

3.1. Mechanical model

The soft actuator can be simplified as a solid cylindrical shaft. The effect of the dentations on the soft actuator is simplified as a transformation of the moment of inertia (I). The effect of the driven cables can be reduced to a concentrated force acting at the end and a distributed force acting along the cable as a result of friction. Therefore, the soft robot working in the horizontal direction can be considered as a root-fixed cantilever beam of length L and bending stiffness $EI(s)$. The curvature of the beam satisfies

$$\rho = \frac{d\varphi}{ds} = \frac{M(s)}{EI}, \tag{1}$$

where ρ is the deformation curvature of a point on the beam, φ is the deformation angle of the beam, $M(s)$ is the bending moment, EI is the bending stiffness, and s is the arc coordinates. Common external and internal loads on actuators can be divided into three categories: external concentrated load, self-gravity load and driving force.

3.1.1. External concentrated load bending. Consider a force F in a constant direction and with a constant magnitude acting at a distance L_F from the root on the soft actuator. The geometric relationship of the soft actuator under this load is shown in figure 4(A), where α is the load azimuth. Differentiating equation (1) yields

$$EI\frac{d^2\varphi}{ds^2} + F\sin(\alpha+\varphi) = 0. \tag{2}$$

The boundary condition can be obtained as

$$\begin{cases} \varphi(0) = & 0 \\ \frac{d\varphi}{ds}(s) = & 0 \quad (s \geqslant L_F) \end{cases}. \tag{3}$$

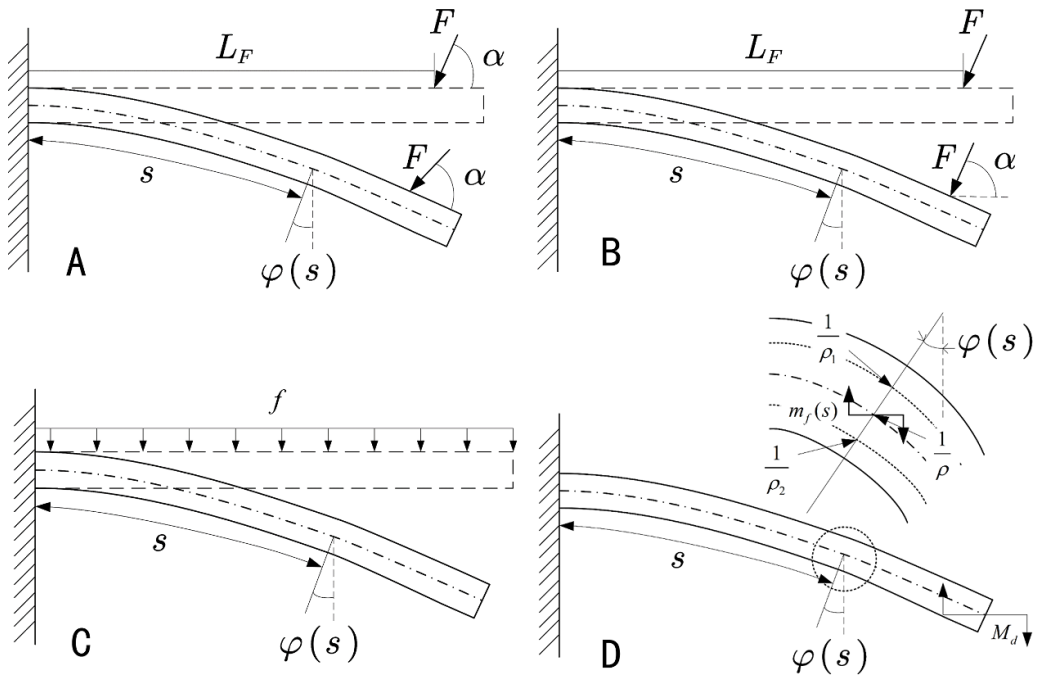

Figure 4. External and internal load. (A) Schematic diagram of bending with directional load. (B) Schematic diagram of bending with follow on load. (C) Schematic diagram of bending with gravity. (D) Schematic diagram of bending with driving force.

Considering the transformation as

$$z(s) = \alpha + \varphi(s), \tag{4}$$

and substituting equation (4) into equations (2) and (3), a new governing equation and boundary condition can be obtained as

$$EI\frac{d^2z}{ds^2} + F\sin(z) = 0, \tag{5}$$

and

$$\begin{cases} z(0) = & \alpha \\ \frac{dz}{ds}(s) = & 0 \quad (s \geqslant L_F) \end{cases}. \tag{6}$$

Integrating equation (5) from 0 to s yields

$$\frac{1}{2}\left(\frac{dz}{ds}\right)^2\Bigg|_0^s = \frac{F}{EI}\cos z|_0^s. \tag{7}$$

Substituting equation (6) into equation (7), we obtain

$$\left(\frac{dz}{ds}\right)^2\Bigg|_0 = \frac{2F}{EI}\left(\cos\alpha - \cos z|_{L_F}\right). \tag{8}$$

Thus, the deflection curve of the manipulator under the directional load is obtained as

$$\left(\frac{dz}{ds}\right)^2\Bigg|_s = \frac{2F}{EI}\left(\cos z|_s - \cos z|_{L_F}\right). \tag{9}$$

Consider a force F at a constant angle in the axial direction and with a constant magnitude acting at a distance L_F from the root on the soft actuator. The geometric relationship of the soft actuator under this load is shown in figure 4(B). Differentiating equation (1) yields the governing equation as

$$EI\frac{d^2\varphi}{ds^2} + F\sin(\alpha + \varphi - \varphi(L)) = 0. \tag{10}$$

The boundary condition can be obtained as

$$\begin{cases} \varphi(0) = & 0 & \\ \frac{d\varphi}{ds}(s) = & 0 & (s \geqslant L_F) \\ \varphi(s) = & \varphi(L_F) & (s \geqslant L_F) \end{cases}. \tag{11}$$

Considering the transformation as

$$z(s) = \alpha + \varphi(s) - \varphi(L_F), \tag{12}$$

and substituting equation (12) into equations (10) and (11), a new governing equation and boundary condition can be obtained as

$$EI\frac{d^2z}{ds^2} + F\sin(z) = 0, \tag{13}$$

and

$$\begin{cases} z(0) & = & \alpha - \varphi(L_F) & \\ \frac{dz}{ds}(s) & = & 0 & (s \geqslant L_F) \end{cases}. \tag{14}$$

This is similar to equations (5) and (6), and the result can also be expressed in the form of equation (9).

3.1.2. Self-gravity bending. Using the geometric relationship of the beam under a uniform load (figure 4(C)), we obtain the following governing equation by differentiating equation (1):

$$\frac{d^3(\varphi)}{ds^3} = \frac{f\cdot\cos(\varphi)}{EI}. \tag{15}$$

The boundary condition can be obtained as

$$\begin{cases} \varphi(0) &= 0 \\ \left.\frac{d\varphi}{ds}\right|_L &= 0 \end{cases}. \tag{16}$$

Substituting the integral of equation (15) into the boundary condition

$$\frac{1}{6}\left(\frac{d\varphi}{ds}\right)^3\Big|_0^s - \frac{1}{2}\left(\frac{d^2\varphi_{(L)}}{ds^2}\right)\cdot s = \frac{f}{EI}\cdot\cos(\varphi_{(s)})|_s^0. \tag{17}$$

We can simplify equation (17) to obtain

$$\frac{1}{2}\left(\frac{d^2\varphi_{(L)}}{ds}\right)^2 = -\frac{1}{6L}\left(\frac{d\varphi}{ds}\right)^3\Big|_0 - \frac{f}{EIL}\left[1-\cos(\varphi(L))\right]. \tag{18}$$

Thus, the deflection curve of the manipulator under the action of gravity is obtained as

$$\frac{1}{6}\left(\frac{d\varphi}{ds}\right)^3\Big|_s + \left(\frac{s}{L}-1\right)\cdot\frac{1}{6}\left(\frac{d\varphi}{ds}\right)^3\Big|_0 + \frac{f}{EI}\left[\frac{s}{L}\left(1-\cos(\varphi(L))\right) - \left(1-\cos(\varphi(s))\right)\right] = 0. \tag{19}$$

As a result, the mechanical model of the manipulator under the action of gravity can be deduced. The static model plays an essential role in the performance analysis, trajectory control, and practical manipulator application.

3.1.3. Driving force. As seen in equation (1), the shape of the soft actuator is influenced by the distribution of the bending moment to which it is subjected. The bending moment of the driving force acting on the soft actuator can be decomposed into two parts: the concentrated moment M_d acting at the end fixed point and the distributed moment $m_f(s)$ from the friction between the drive line and the via-hole. These components satisfy

$$F\cdot d = M_d + \int_0^L m_f(s)ds, \tag{20}$$

where F denotes the total driving force and d denotes the distance between the two driving cables.

When the soft actuator is stationary, the direction and magnitude of the friction force and its distribution depend on the previous state. If the previous state of the soft actuator is not available an error band can be estimated using the maximum static friction force along the same direction. The shape of the soft actuator then falls within this error band. The concentrated moment M_d acting on the end fixed point is considered separately. According to equation (1), the curvature ρ is a constant. Thus if the friction between the driving line and the via-hole can be neglected, the shape of the soft actuator bending under only the driving force should be a segment arc. We now consider the effect of the maximum static friction. Because the friction force is related to the normal pressure, the following relationship can be obtained:

$$m_f = \mu d\cdot T(s)\cdot\rho(s), \tag{21}$$

where μ denotes the maximum static friction coefficient and $T(s)$ denotes the tensile force on the drive line at s. From equation (20), we have that

$$T(s) = F - \int_0^s T(l)\,\rho(l)\,dl. \tag{22}$$

Differentiating equation (22) gives

$$T'(s) = -T(s)\,\rho(s). \tag{23}$$

Solving equation (23) under the condition that $T(0) = F$, we find that

$$T(s) = F\cdot\exp\left(-\int_0^s \rho(l)\,dl\right). \tag{24}$$

Equation (21) can then be written as

$$m_f = \mu dF\cdot\exp\left(-\int_0^s \rho(l)\,dl\right)\cdot\rho(s). \tag{25}$$

Substituting equation (25) into equation (1), the drive equation considering only the frictional drive is

$$\frac{\int_s^L \mu dF\cdot\exp\left(-\int_0^l \rho(l)\,dl\right)\cdot\rho(l)\;dl}{EI} = \rho(s). \tag{26}$$

Using the simplification $\rho = \frac{d\varphi}{ds}$, equation (26) can be formulated as

$$\frac{\mu dF\cdot\exp(-\varphi(l))|_{l=s}^{l=L}}{EI} = \frac{d\varphi}{ds}. \tag{27}$$

Integrating the above equation gives

$$\exp\left(\frac{\mu dF}{EI}\cdot\frac{s}{\exp(\varphi(L))}\right) = \frac{\exp(\varphi(L)) - \exp(\varphi(s))}{\exp(\varphi(L)) - 1}. \tag{28}$$

According to equation (1), when analyzing the trajectory of the final soft actuator, the bending effect caused by the load and the concentrated moment acting on the root is first calculated. Because equation (25) only considers the maximum static friction that hinders the deformation, the obtained trajectory will have a unidirectional deviation. The effect of friction will increase when the driving force is greater, but the direction of

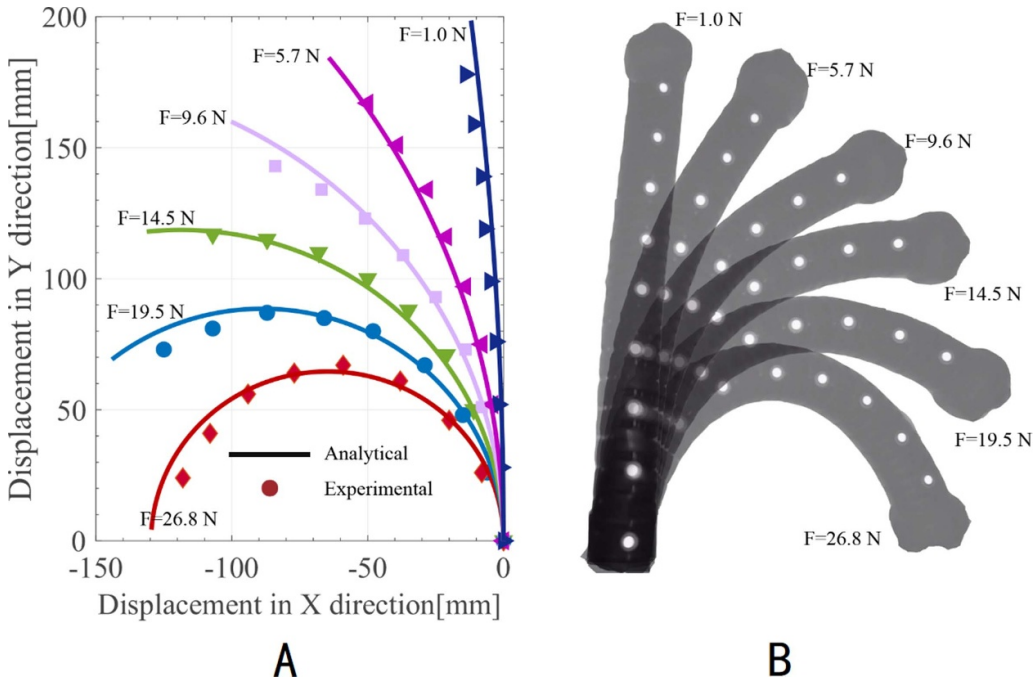

Figure 5. Comparison of experimental result with analysis. (A) Bending of the soft actuator under different driving forces compared with the trajectory calculated by the analysis method. (B) Photograph of soft actuator bending under different driving forces.

the friction can be determined by the trajectory before deformation. In addition, because the soft actuator is an underdriven system, the same drive configuration can form different trajectories under different external loads. When changing between trajectories, the main factors that hinder this change are the stiffness of the deformer itself and the friction between the drive line and the via-hole.

3.2. Mechanical performances

The model described above can be used to estimate the static trajectory of the soft robotic actuator under a driving force. To verify the validity of this model, a series of experiments are conducted using a soft actuator with a full length of $L = 200\,\text{mm}$, bending stiffness of $EI = 1.0 \times 10^4\,\text{N}\cdot\text{mm}^2$ and drive cable spacing of $d = 8\,\text{mm}$. The friction coefficient between the drive cable and the via-hole is $\mu = 0.1$. The soft actuator is bent into different trajectories under different driving forces. The trajectories of the soft actuator under these driving forces are also calculated separately, as shown in figure 5. The trajectories shown in the figure account for the maximum static friction that impedes the motion. This is closer to the actual situation, but the overall result has a small one-way deviation. This means that, in practice the friction between the drive cable and the via-hole is closer to the maximum static friction in one direction, but at rest the friction between the two is less than the maximum static friction. Thus, for a certain trajectory, the current trajectory is maintained differently in each direction. Coupled with the fact that soft actuators are often used in non-structural environments that do not require particularly precise control, the friction can be simplified to a one-way maximum friction in practical applications. The soft actuator design used in this study features grooves aligned in the same direction as the drive lines, with each pair of drive lines orthogonal to one another. This configuration ensures that the motion between the two pairs of drive lines remains uncoupled. Thus, the proposed model can be applied to multi-axis motion with comparable results.

4. Adaptive neural control

An adaptive neural controller that can converge to a prescribed error limit in finite time for model-free control has previously been designed [39]. This study combines this adaptive neural controller with the quasi-static model described above to provide an effective soft-actuator control method that does not depend on the dynamics model. We also account for the flexible 3D printing material being highly viscous, which can reduce the RBFNN fitting accuracy. The whole controller structure is shown in figure 6. The controller mainly consists of seven parts: OpenCV-based target extraction, RBFNN, Euler–Bernoulli beam model, intermediate variable, coordinate transformation, adaptation law and driving force controller. The OpenCV-based target point velocity extraction module achieves the extraction of the velocity of the target point from the image stream returned by the camera using binarization of the image feedback, and extracting the center coordinates of the elliptical highlighted area of the target point in the image. The RBFNN is used to fit the dynamic model of the soft actuator using the feedback driving force, velocity and estimated position. The Euler–Bernoulli beam model is

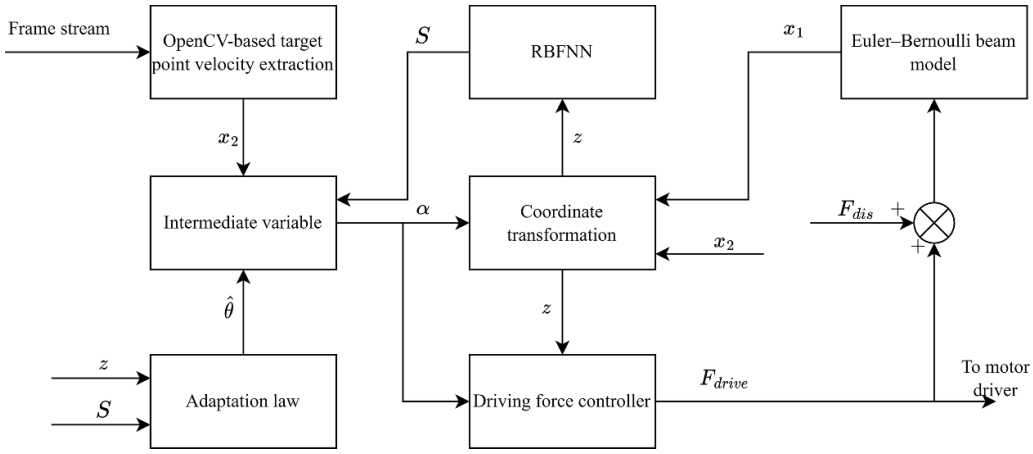

Figure 6. Control flowchart of the proposed control scheme.

used to estimate the steady-state position of the soft actuator from the sensor feedback drive, preventing the position delay caused by viscoelasticity from slowing the RBFNN convergence. Intermediate variables are used to calculate the virtual control law, while the coordinate transformation uses the auxiliary system to transform the input speed and position. The driving force controller uses the virtual control law and the current soft actuator state to calculate the driving force and output it to the motor drive.

Given the orthogonal arrangement of the drive cables in the soft actuator utilized in this study, no inter-cable coupling exists. The system function of the soft actuator satisfies the following relations:

$$\begin{cases} \dot{x}_1(t) = \varphi_{1j}(x(t))\, x_2(t) + \phi_{1j}(x(t)) \\ \dot{x}_2(t) = \varphi_{2j}(x(t))\, u_j(t) + \phi_{2j}(x(t)) \\ y(t) \;\; = \qquad\qquad x_1(t) \end{cases}, \tag{29}$$

where $x(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^T \in \mathbb{R}^n$ are the state variable and system output, respectively. $u_j(t) \in \mathbb{R}$ is the input corresponding to the jth drive cable, and $\varphi_{ij}(\cdot)$, $\phi_{ij}(\cdot)$ are unknown smooth nonlinear functions. Because φ is related to the mass and damping distribution of the soft actuator, the two known constants $\underline{\varphi}_i$ and $\bar{\varphi}_i$ satisfy $0 < \underline{\varphi}_i < \varphi_{ij}(\cdot) < \bar{\varphi}_i$. To facilitate the description, the input $x(t)$ is transformed according to

$$z_1 = \frac{e_h^2 e}{e_h^2 - e^2} \tag{30}$$

$$z_2 = x_2 - \alpha_1, \tag{31}$$

where $e(t) = y(t) - y_r(t)$ is the tracking error. The tracking restraint condition $e_h(t)$ should satisfy $-e_h(t) < e(t) < e_h(t)$. α is the virtual control law and $y_r(t)$ is the tracking result. For equation (29), the system functions are unknown and cannot be directly used to design the desired controller. Thus, the RBFNN is applied to approximate an arbitrary unknown nonlinear function. For a given accuracy $\tau > 0$, with sufficiently large node number l, the RBFNN can approximate any continuous function $F(Z)$ over a compact set Ω_Z such that

$$F_i(Z) = W_i^T S_i(Z) + \delta(Z), |\delta(Z)| \leqslant \tau, \tag{32}$$

where $Z \in \Omega_Z \subseteq \mathbb{R}^q$ represents the input vector, $W_i \in \mathbb{R}^l$ denotes the weight vector, $l > 1$ is the node number of the neural network, $\delta(Z)$ is an approximation error, and $S_i(Z) = [s_1(Z), \ldots, s_l(Z)]^T$ is the basis function vector with $s_i(Z)$ generally set to a Gaussian function

$$s_i(Z) = \exp\left[\frac{-(Z-\upsilon_i)^T(Z-\upsilon_i)}{\eta^2}\right], i = 1, 2, \ldots, l_k, \tag{33}$$

where $\nu = [\nu_{i1}, \nu_{i2}, \ldots, \nu_{iq}]^T$ and η represent the center of the receptive field and the width of the Gaussian function, respectively. $F_i(Z)$ in equation (32) is the dynamic model to be fitted, and this work uses two RBFNNs for the soft actuator state-dependent $X_1 = [x_1, y_r, \dot{y}_r]^T$, $X_2 = \left[x_1, x_2, y_r, \dot{y}_r, \lambda, \hat{\theta}_1\right]^T$ as the input vector, but the ideal weight matrix W_i is unknown and the adaptive parameters are designed as $\theta_i = |W_i|^2$, and $\hat{\theta}$ is used as the estimate of θ for the computation of the driving parameters. W is an unknown ideal constant weight vector. Therefore, adaptive parameters are designed as $\theta_i = \|W_i\|^2$. The first adaptation law is designed as

$$\dot{\hat{\theta}}_1 = \frac{r_1}{2l_1}(z_1\lambda)^2 S_1^T(X_1) S_1(X_1) - \sigma_1\hat{\theta}_1 - \frac{b_1\hat{\theta}_1^3}{r_1}, \tag{34}$$

where $b1, \sigma, r1$ are the positive design parameters, and $\lambda = \frac{\partial z_1}{\partial e}$. l_i is the number of nodes in neural network S_i. The first virtual control signal is designed as

$$\alpha_1 = -\frac{z_1\tilde{\alpha}_1^2}{\underline{\varphi}_1\lambda\sqrt{z_1^2\tilde{\alpha}_1^2 + \varepsilon_1^2}}, \tag{35}$$

$$\tilde{\alpha}_1 = K_{11}\left(\frac{1}{2}\right)^{\frac{3}{4}}\frac{\left(z_1^2\right)^{\frac{3}{4}}}{z_1} + K_{12}\left(\frac{1}{2}\right)^2 z_1^3 + \frac{z_1\lambda^2}{2l_1}\hat{\theta}_1 S_1^T(X_1) S_1(X_1) + \Psi, \tag{36}$$

where $\varepsilon_1, K_{11}, K_{12}$ are the positive design parameters, and $\Psi = \frac{\partial z_1}{\partial e_h}\dot{e}_h$. For the system described by equation (29), a second

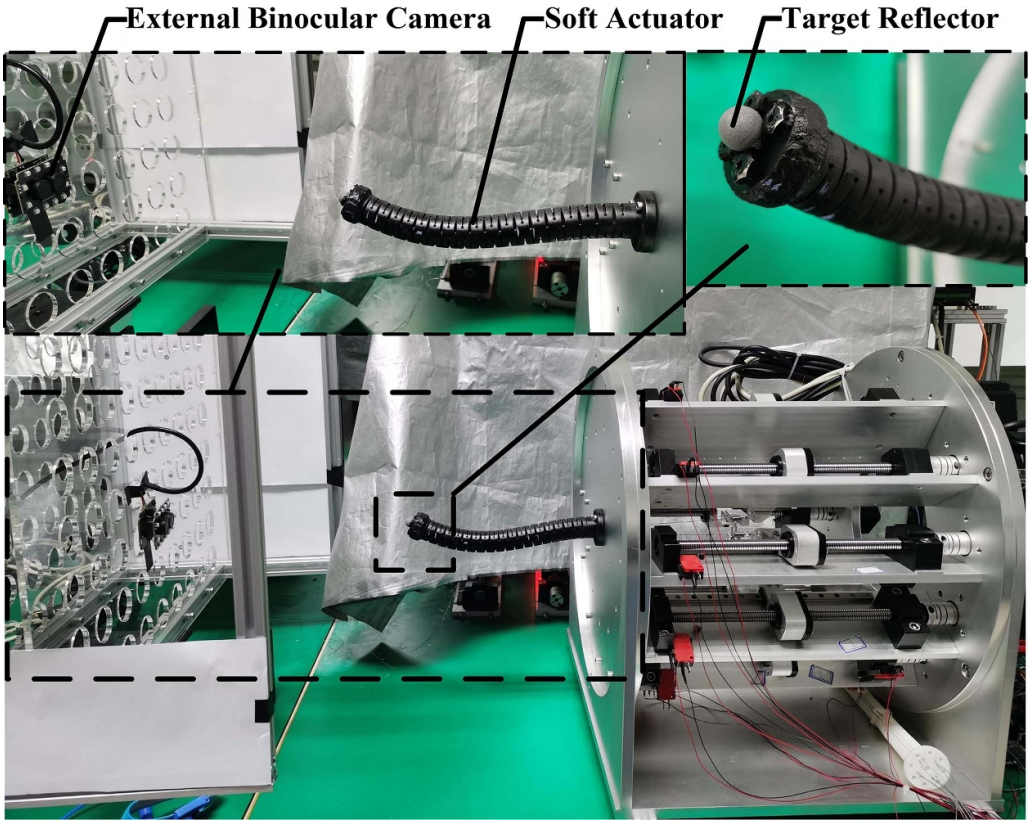

Figure 7. Experimental system for a soft actuator.

virtual control signal and adaptation law are designed as

$$\tilde{\alpha}_2 = K_{21}\left(\frac{1}{2}\right)^{\frac{3}{4}}\frac{\left(z_2^2\right)^{\frac{3}{4}}}{z_2} + K_{22}\left(\frac{1}{2}\right)^{2} z_2^3 + \frac{z_2}{2l_2}\hat{\theta}_2 S_2^T\left(X_2\right) S_2\left(X_2\right), \tag{37}$$

$$\dot{\hat{\theta}}_2 = \frac{r_2}{2l_2} z_2^2 S_2^T\left(X_2\right) S_2\left(X_2\right) - \sigma_2\hat{\theta}_2 - \frac{b_2\hat{\theta}_2^3}{r_2}. \tag{38}$$

According to equation (35) the driving force controller of *j*th cable can be designed as

$$u_j = -\frac{z_2\tilde{\alpha}_2^2}{\underline{\varphi}_2\sqrt{z_2^2\tilde{\alpha}_2^2 + \varepsilon_2^2}}. \tag{39}$$

5. Experimental results and discussions

This section reports the result from a series of experiments to validate the effectiveness of the designed control scheme. The soft actuator and its drive system are as described in section 2. The experimental system is shown in figure 7. An external binocular camera (1920 × 1080 @ 60 FPS for each camera) and a target reflector for tracking are added to the soft actuator and drive system. The binocular camera is directly connected to the computer and the position of the target reflector is extracted by OpenCV. The controller runs under the Simulink toolbox of MATLAB on a computer terminal, and the driving force can be calculated from the feedback target reflector position and the preset target trajectory. The calculated driving force is transmitted to the MCU(STM32F103VET6) via the USART serial port to control the motor motion and the driving force on the drive cable is sent back to the computer by the MCU. The communication frequency is about 100 Hz. The improved control method in this paper, the control method proposed in [39], and the conventional PID method are tested. The three methods are used to test tracking exponential functions and sine respectively.

5.1. Exponential tracking experiment

Figure 7 shows the initial zero position of the soft actuator. The trajectory is set to $y_r(t) = -0.075 + 0.1\exp(-t) - 0.025\exp(-4t)$, and the tracking restraint condition is assigned as $e_h = 0.005\exp(-t) + 0.002$. The test considers single-DoF motion. A pair of drive cables is controlled to drive the vertical motion, so that the projection of the target reflector in the vertical direction tracks the given trajectory. The controller output driving force is that on the upper drive cable minus that on the lower drive cable. If the controller output force is positive, the lower drive cable is assigned the default tension of 0.2 N; if the controller output force is negative, the upper drive cable is assigned the default driving force of 0.2 N. The three experiments were

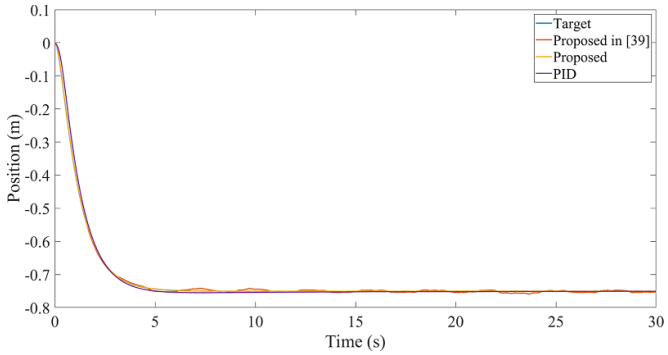

Figure 8. Tracking position of the actuator with exponential signal.

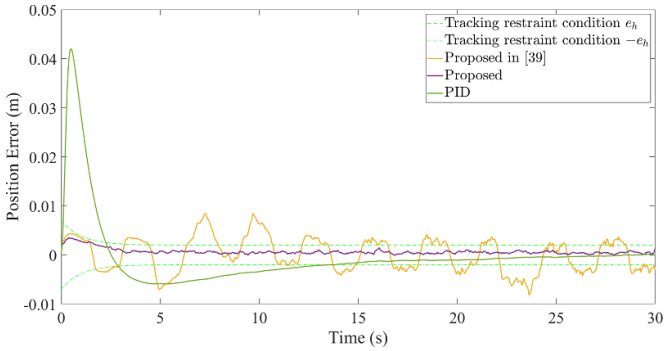

Figure 9. Tracking position error of the actuator with exponential signal.

performed using are tracking the same trajectory using the method proposed in [39], the modified method of this paper and the traditional PID method. The parameters of the controller are set to $K_{11} = K_{12} = K_{21} = K_{22} = 0.1, \varepsilon_1 = \varepsilon_2 = 1, r_1 = r_2 = \sigma_1 = \sigma_2 = b_1 = b_2 = 1$. The initial conditions are $\hat{\theta}(0) = [0.2, 0.1]^T$.

The experimental results are presented in figures 8–10. Figure 8 shows the tracking trajectories under the three control methods. The tracking speed of proportion-integral-differential (PID) controller is weaker than the other two methods, but the error range is smaller than that of the method from [39] when reaching the stable position of the target. Figure 9 shows the position tracking error for the controller experiments based on the three methods. When quasi-static model feedback is not used, the control errors all fail to converge to the given tracking constraints, indicating that the controller has failed. However, using the control method proposed in this paper, the tracking error is smaller than the specified tracking constraint for the entire process, demonstrating that this method effectively improves the convergence of the error. Figure 10 gives the measured driving force of the controller experiments based on the three methods. These prescribed boundary-constrained control methods require higher drive force control performance for fast convergence. In contrast, the drive force of the PID method fluctuates much less around the stable poses. While the drive force required by the controller proposed in [39] fluctuates more throughout. This shows that the controller proposed in this paper can achieve better control results when used to soft actuators. It effectively eliminates jitter in the control of soft actuators by predefined boundary constraint controllers with direct position feedback. In practice, soft actuators have lower driving force and are more prone to force saturation. Therefore, the adaptive tracking control with prescribed boundary constraints in [39] is not suitable for soft actuators. However, a quasi-static model of the soft actuator can be applied to allow the control error to converge.

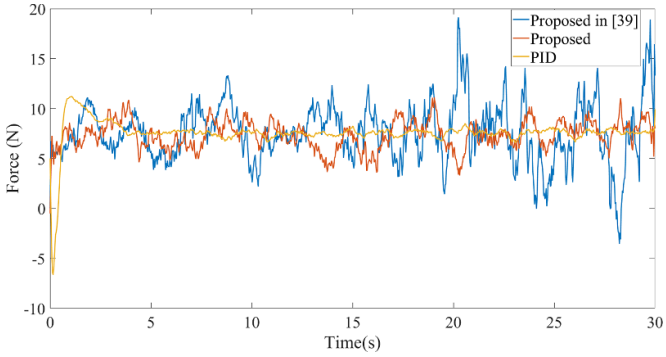

Figure 10. Driving force of the actuator with exponential signal.

5.2. Sine tracking experiment

In this experiment, the trajectory is set to $y_r(t) = -0.1\cos\left(\frac{\pi}{15}t\right) + 0.1$, and the tracking restraint condition is assigned as $e_h = 0.005\exp(-t) + 0.002$. The remaining parameters are kept the same as in the exponential tracking experiment.

The experimental results are presented in figures 11–13. Figure 11 illustrates the position results of the three control methods utilized to track the sine wave signal. The PID method experiences difficulty in accurately tracking the target signal. On the other hand, the two prescribed boundary-constrained control methods can effectively track the target trajectory. Notably, the proposed method displays less fluctuation near the target. Figure 12 displays the error results of the three control methods when tracking the sine signal. Only the control method proposed in this paper achieves the prescribed boundary-constrained. The method proposed in [39] exhibits fluctuations in the error beyond the limit, indicating control method failure. Figure 13 portrays the results of driving force for each of the three control methods tracking the sine signal. The PID method experiences the smallest fluctuation of the driving force, followed by the method proposed in this paper. In contrast, the method proposed in [39] experiences the largest fluctuation. This finding further confirms that the improved method in this paper successfully resolves the problem that the method proposed in [39] fails to address when utilized for soft actuators, while retaining the characteristics and dynamic performance of the prescribed boundary-constrained.

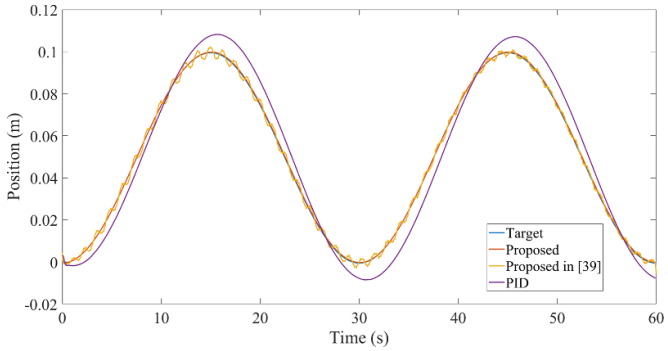

Figure 11. Tracking position of the actuator with sine signal.

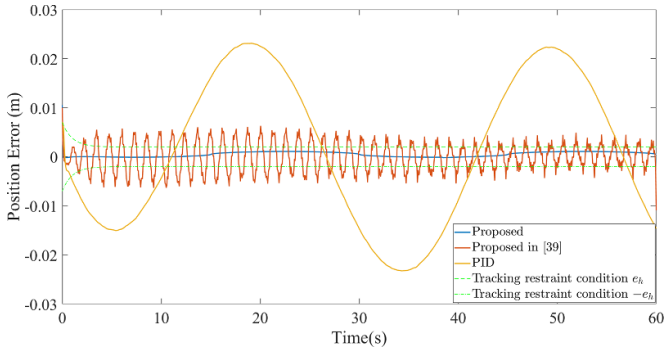

Figure 12. Tracking position error of the actuator with sine signal.

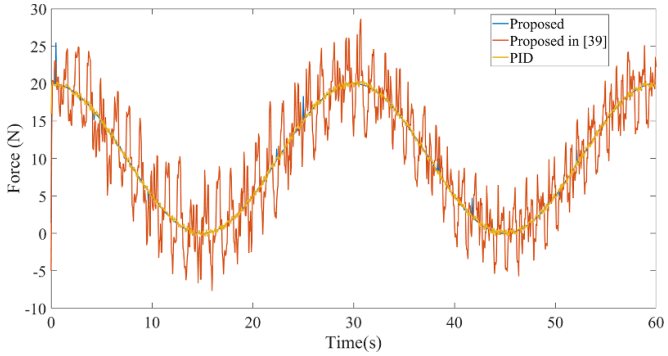

Figure 13. Driving force of the actuator with sine signal.

5.3. Discussions

In general, this paper proposes an adaptive tracking controller with position feedback using a quasi-static model, which can be applied to soft actuators. The proposed method not only fully harnesses the actuator's performance, thereby achieving faster error convergence compared to the PID method, but also effectively maintains the error within prescribed boundaries during the convergence process. Additionally, this method addresses the convergence failure encountered in soft actuators, as the method described in [39]. As the quasi-static model is relatively simple, the complexity of the control system is not significantly increased. Nevertheless, the accuracy of the quasi-static model implies that control results may be impacted by steady-state errors. However, as illustrated in figures 9 and 12, the magnitude of such errors remains within the prescribed boundary constraint, thereby not affecting the practical application of the actuator.

6. Conclusions

This paper has described a control method for 3D-printed cable-driven soft actuators. As the 3D-printed flexible material is viscous, adaptive neural controllers are ineffective. Thus, a new control method based on Euler–Bernoulli model feedback has been developed. The proposed method achieves trajectory tracking of the 3D-printed cable-driven soft actuator by establishing a quasi-static model with a predefined boundary-constrained adaptive neural controller. The method uses Euler–Bernoulli quasi-static model feedback along with a neural network technique to fit the dynamics of the actuator and constrain the actuator trajectory error within predefined bounds. Experimental results show that the tracking error of the soft actuator converges to within the predefined constraints. The potential of the predefined boundary control theory for the control of soft actuators has been demonstrated. In future work, we will consider the design of control methods for various flexible actuators made by 3D printing with prescribed performance constraints that match their characteristics. To leverage the flexibility advantages of soft robots, efforts are underway to design and test soft actuators with higher aspect ratios.

Data availability statement

The data cannot be made publicly available upon publication because no suitable repository exists for hosting data in this field of study. The data that support the findings of this study are available upon reasonable request from the authors.

Funding

Financial supports from the National Natural Science Foundation of China (Grant No. 11 972 343) and Jilin Province Science and Technology Research Project (Grant No. 20 200 40 4149YY).

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

ORCID iD

Zhenbang Xu https://orcid.org/0000-0002-4120-2448

References

[1] Zhou X, Majidi C and O'Reilly O M 2015 *Int. J. Solids Struct.* **64** 155
[2] Chen G, Yang X, Xu Y, Lu Y and Hu H 2022 *Smart Mater. Struct.* **32** 015004

[3] Rus D and Tolley M T 2015 *Nature* **521** 467
[4] TolleyMichael T, Onal C D and Rus D 2014 *Soft Robot.* **1** 1
[5] Bartlett N W, Tolley M T, Overvelde J T, Weaver J C, Mosadegh B, Bertoldi K, Whitesides G M and Wood R J 2015 *Science* **349** 161
[6] Chen L, Yang C, Wang H, Branson D T, Dai J S and Kang R 2018 *Mech. Mach. Theory* **130** 109
[7] Li Z and Du R 2013 *Int. J. Adv. Robot. Syst.* **10** 209
[8] Gu G, Zou J, Zhao R, Zhao X and Zhu X 2018 *Sci. Robot.* **3** 25
[9] Lee C, Kim M, Kim Y J, Hong N, Ryu S, Kim H J and Kim S 2017 *Int. J. Control Autom. Syst.* **15** 3
[10] Duduta M, Clarke D R and Wood R J 2017 *2017 IEEE Int. Conf. on Robotics and Automation (ICRA)* (IEEE) pp 4346–51
[11] Li W-B, Zhang W-M, Zou H-X, Peng Z-K and Meng G 2018 *IEEE/ASME Trans. Mechatronics* **23** 1630
[12] Zhakypov Z, Heremans F, Billard A and Paik J 2018 *IEEE Robot. Autom. Lett.* **3** 2894
[13] Li T *et al* 2017 *Sci. Adv.* **3** e1602045
[14] Zhao H, Li Y, Elsamadisi A and Shepherd R 2015 *Extreme Mech. Lett.* **3** 89
[15] Pagoli A, Chapelle F, Corrales-Ramon J-A, Mezouar Y and Lapusta Y 2021 *Smart Mater. Struct.* **31** 013001
[16] Chen G, Yang X, Zhang X and Hu H 2021 *Appl. Ocean Res.* **109** 102551
[17] Liang X, Cheong H, Sun Y, Guo J, Chui C K and Yeow C-H 2018 *IEEE Robot. Autom. Lett.* **3** 2702
[18] Wang H, Zhang R, Chen W, Wang X and Pfeifer R 2017 *Surgical Endosc.* **31** 3152
[19] Deng T, Wang H, Chen W, Wang X and Pfeifer R 2013 *2013 IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)* (IEEE) pp 728–33
[20] Wang H, Wang C, Chen W, Liang X and Liu Y 2016 *IEEE/ASME Trans. Mechatronics* **22** 18
[21] George Thuruthel T, Ansari Y, Falotico E and Laschi C 2018 *Soft Robot.* **5** 149
[22] Duriez C 2013 *2013 IEEE Int. Conf. on Robotics and Automation* (IEEE) pp 3982–7
[23] Zhang Z, Dequidt J, Kruszewski A, Largilliere F and Duriez C 2016 *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)* (IEEE) pp 5509–14
[24] Till J, Aloi V and Rucker C 2019 *The Int. J. Robot. Res.* **38** 723
[25] Lin P T, Shahabi E, Yang K-A, Yao Y-T and Kuo C-H 2019 *Iftomm World Congress on Mechanism and Machine Science* (Berlin: Springer) pp 617–25
[26] Fan J, Wang S, Yu Q and Zhu Y 2020 *Soft Robot.* **7** 615
[27] Rucker D C and Webster III R J 2011 *IEEE Trans. Robot.* **27** 1033
[28] Yuan H, Zhou L and Xu W 2019 *Mech. Mach. Theory* **135** 130
[29] Kim D *et al* 2021 *PLoS One* **16** e0246102
[30] Yip M C and Camarillo D B 2014 *IEEE Trans. Robot.* **30** 880
[31] Chen G, Lu Y, Yang X and Hu H 2022 *Robot. Auton. Syst.* **154** 104116
[32] Wang R, Zhang C, Zhang Y, Tan W, Chen W and Liu L 2022 *Adv. Mater. Technol.* **8** 2200962
[33] Sai H, Xu Z, Xia C and Sun X 2022 *Nonlinear Dyn.* **110** 431
[34] Gao Y-F, Sun X-M, Wen C and Wang W 2016 *IEEE Trans. Neural Netw. Learn. Syst.* **28** 1520
[35] Sai H, Li Y, He S, Zhang E, Zhu M and Xu Z 2022 *Proc. Inst. Mech. Eng.* C 09544062221139968
[36] Bechlioulis C P and Rovithakis G A 2009 *Automatica* **45** 532
[37] Bechlioulis C P and Rovithakis G A 2010 *IEEE Trans. Autom. Control* **55** 1220
[38] Ma J, Xu S, Li Y, Chu Y and Zhang Z 2018 *J. Franklin Inst.* **355** 5503
[39] Zhang X, Tan J, Yao Y and Wu J 2022 *Int. J. Adapt. Control Signal Process.* **36** 653