

# Dynamic Control of Multisection Three-Dimensional Continuum Manipulators Based on Virtual Discrete-Jointed Robot Models

Chengshi Wang<sup>✉</sup>, Chase G. Frazelle, John R. Wagner<sup>✉</sup>, Senior Member, IEEE,  
and Ian D. Walker<sup>✉</sup>, Fellow, IEEE

**Abstract**—Despite the rise of development in continuum manipulator technology and application, a model-based feedback closed-loop control appropriate for continuum robot designs has remained a significant challenge. Complicated by the soft and flexible nature of the manipulator body, control of continuum structures with infinite dimensions proves to be difficult due to their complex dynamics. In this article, a novel strategy is designed for trajectory control of a multisection continuum robot in three-dimensional space to achieve accurate orientation, curvature, and section length tracking. The formulation connects the continuum manipulator dynamic behavior to a virtual discrete-jointed robot whose degrees of freedom are directly mapped to those of a continuum robot section under the hypothesis of constant curvature. Based on this connection, a computed torque control architecture is developed for the virtual robot, for which kinematics and dynamic equations are constructed and exploited, with appropriate transformations developed for implementation on the continuum robot. The control algorithm is implemented on a six degree-of-freedom two-section OctArm continuum manipulator. Experiments show that the proposed method could manage simultaneous extension/contraction, bending, and torsion actions on multisection continuum robots with decent performance (arc length and curvature error of  $\pm 4$  mm and  $\pm 0.35$  m $^{-1}$ ). The designed dynamic controller can reduce the curvature tracking error and rise time by up to 48.1% and 94.8% compared to the traditional proportional-integral-derivative controller during two-section maneuvers.

**Index Terms**—Continuum robot, forward kinematics, inverse kinematics, manipulator dynamics, motion control.

Manuscript received March 18, 2020; revised May 8, 2020; accepted May 31, 2020. Date of publication June 3, 2020; date of current version April 15, 2021. This work was supported in part by the U.S. National Science Foundation under Grant IIS-1527165 and Grant IIS-1718075, in part by the NASA under contract NNX12AM01G, and in part by the NASA Space Technology Research Fellowship under contract 80NSSC17K0173. Recommended by Technical Editor P. Ben-Tzvi and Senior Editor H. Qiao. (*Corresponding author: Chengshi Wang.*)

Chengshi Wang and John R. Wagner are with the Department of Mechanical Engineering, Clemson University, Clemson, SC 29634 USA (e-mail: chengsw@clemson.edu; jwagner@clemson.edu).

Chase G. Frazelle and Ian D. Walker are with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634 USA (e-mail: cfrazel@clemson.edu; iwalker@clemson.edu).

Color versions of one or more of the figures in this article are available online at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMECH.2020.2999847

## I. INTRODUCTION

A CONTINUUM robot [1] is a bioinspired slender hyper-redundant manipulator with theoretically infinite degrees-of-freedom (DoF), which provides remarkable capabilities for reach, manipulation, and dexterity in a cluttered environment. Recent progress toward biological systems has enabled continuum robotics research to expand rapidly, promising to extend the use of continuum robots into many new environments and providing them with capabilities beyond the scope of their rigid-link counterparts [2], [3]. Long-term success for the practical application of continuum robots heavily relies on the development of real-time controllers that deliver accurate, reliable, and energy-efficient control. However, the development of high-performance model-based control strategies for continuum robots proves to be challenging due to multiple reasons. Since the manipulators must be modeled as continuous curves, the kinematic and dynamic models are difficult to derive. Also, control of continuum structures is complicated by the intrinsic underactuated nature of the backbone. Additionally, noncollocation of actuators with configuration space variables impedes the closed-loop control design (error calculation) in either space. The current scarcity of suitable local sensor technologies also contributes to the difficulty of direct internal sensing of backbone shape, which, in turn, hinders the control of continuum structures [4].

Over the years, extensive model-based static control strategies have been proposed to design better and more reliable controllers for the continuum robots. Camarillo *et al.* [5], [6] proposed a closed loop configuration and task space controller for tendon-driven continuum manipulators, experimentally validated with a five-DoF per section kinematic model. Bajo *et al.* [7] realized a configuration space controller, which utilizes extrinsic sensory information about configuration and intrinsic sensory information about joint space. An adaptation of the classic hybrid motion/force controller for continuum robots was presented and evaluated in both [8] and [9]. More complex kinematic formulations for continuum robots such as variable constant curvature (VCC) approximation were used by Mahl *et al.* [10] for kinematic control of three-section manipulator with a gripper. Wang *et al.* [11] presented visual servo control of a 2-D image feature point in 3-D space using the VCC model for a cable-driven soft conical manipulator. Conrad *et al.* [12]

applied a closed-loop task space controller on an interleaved continuum-rigid manipulator. Marchese and Rus [13] achieved kinematic control of a pneumatic-actuated soft manipulator that is made from low durometer elastomer. However, static control strategies rely on the steady-state assumption, which impedes fast and accurate motion of continuum robots.

Widely considered the most challenging field in the control of continuum robots [14], model-based dynamic controllers that consider the complete kinematics and dynamics of the whole manipulator have been explored by previous researchers. Gravagne and Walker [15] validated feedforward and feedback proportional-derivative controller through simulations of a planar single multisegment continuum robot. Falkenhahn *et al.* [16] implemented optimal control strategies that consider both the mechanical dynamics and the pressure dynamics to achieve trajectory optimization of continuum manipulators. Marchese *et al.* [17] described a trajectory optimization scheme and dynamic model for a soft planar elastomer manipulator. Falkenhahn *et al.* [18] developed a dynamic controller in actuator space that provides actuator decoupling in combination with feedforward and feedback strategies. Controllers that consider the dynamic behavior of the continuum manipulators are faster, dexterous, efficient, smoother tracking than static controllers.

In this article, we propose a model-based dynamic feedback control architecture that has been specifically designed for controlling continuum robots, extending our previous work [19] not only from two to three dimensions but also from one to two sections. The novel approach to continuum robot control discussed in this article is motivated by reducing computational complexity using a virtual, conventional rigid link robot with discrete joints. The control strategy is developed in the virtual robot coordinates, taking advantage of the well-understood nature of conventional robot dynamics. The virtual robot is selected such that its degrees of freedom are directly mapped to those of the real continuum robot for which control is desired.

Specifically, the abovementioned approach is validated from model development to hardware implementation for control of a multisegment spatial continuum robot. The continuum robot is approximated as a serial rigid-link Revolute-Revolute-Prismatic-Revolute (RRPR) joint spatial robot with an out-of-plane rotation, two in-plane rotations, and a translation in the same plane to create a 3-D virtual rigid-link robot [4]. A block diagram of the joint space dynamic controller by feedback linearization for closed-loop configuration space control is depicted in Fig. 1. The control attempts to achieve tracking of configuration space variables (arc-length  $s$ , curvature  $\kappa$ , and rotational orientation  $\phi$ ) using a computed torque controller in the joint space to calculate virtual torques that are translated into pneumatic pressure in the actuator space. The task space to joint space inverse kinematics are obtained via a desired virtual joint vector, which forms the error vector with the actual virtual joint space variables derived from the continuum robot configuration space. The approach mentioned previously assumes on the constant curvature (CC) approximation [20] for the configuration space model, which means that the continuum robot curvature has the same value at every point on the backbone for a given section.

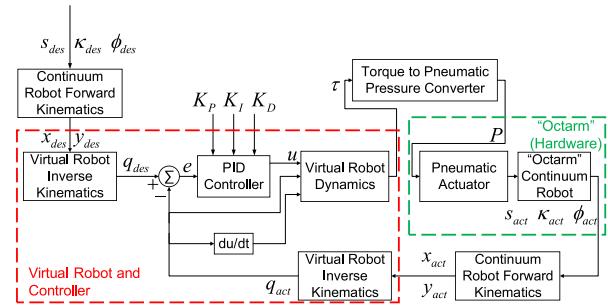
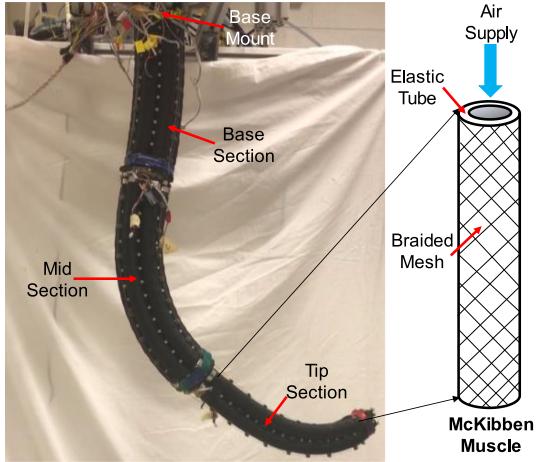


Fig. 1. Block Diagram for Continuum robot control based on virtual robot models.

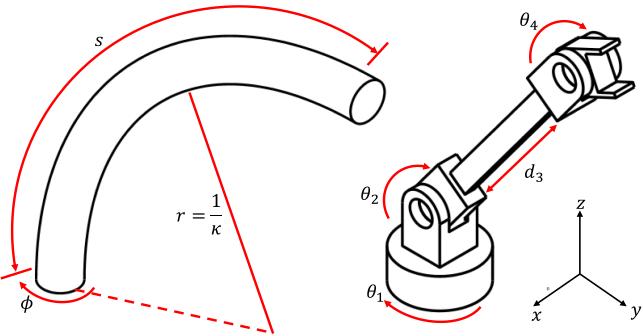
Previous research has also examined the possibility of controlling a continuum structure via exploiting a “virtual” rigid link robot model. However, the control methodologies were either applied only to the bending of a planar continuum section [21], or formulated using a under-parameterized model, which involves merely bending and twisting without considering continuum robot extension/contraction [22], [23]. Prior works, which consider bending [24], twisting, and extension/contraction [25], [26] emphasize dynamic modeling but not control. In contrast, the work in this article is the first attempt to accomplish “virtual” rigid-link robot model-based 3-D control of continuum robots whose configuration space is parameterized by arc-length  $s$ , curvature  $\kappa$ , and rotational orientation  $\phi$ . Such comprehensive parameterization accounts for simultaneous extension/contraction, bending, and torsion actions of continuum robots, therefore matching the controllable range while approximating the motion complexity of continuum robots. In addition, we extend model-based dynamic control research through the application of the computed torque approach that provides virtual rigid link robot dynamics decoupling for the control of multisegment continuum robots.

This virtual discrete-jointed robot model-based controller for continuum manipulators is developed and applied and tested on the octopus biology inspired OctArm [27], a three-section intrinsically pneumatic-actuated continuum manipulator with nine DoF (depicted in Fig. 2). Each OctArm section is constructed using pneumatic “McKibben” muscle extensors [28] with three control channels per section that provide bending, extension, and 360° of 3-D rotation. A closed-loop task space dynamic control [29] and a closed-loop configuration space control using a sliding mode control (SMC) [30], [31] were implemented on the OctArm. More recently, an adaptation-based nonlinear control strategy was adopted for the OctArm [32], but until now, a dynamic controller based on virtual discrete-jointed robot model proposed in this article remains undeveloped for the OctArm.

The rest of this article is organized as follows. The mathematical forward and inverse kinematics, as well as the dynamics model, is discussed in Section II. The control system design is contained in Section III. The experimental methods, as well as one and two section OctArm experimental results, are discussed in Section IV. Finally, Section V concludes this article.



**Fig. 2.** Dynamically controlled OctArm, a pneumatically actuated continuum manipulator with three sections, following a desired configuration trajectory.



**Fig. 3.** Continuum robot section (left), is modeled based on virtual RRPR discrete-jointed robot (right).

## II. MATHEMATICAL MODEL

### A. One-Section Continuum Robot Model

1) *Spatial Continuum Robot Forward Kinematics:* In order to comprehend the continuum robot movements, a forward kinematic model, which relates the configuration space (backbone shape) variables and task (e.g., tip) space variables, needs to be constructed. Such a model lays the foundation for designing control algorithms and is vital for the practical implementation of continuum robot hardware.

The approach to continuum robot forward kinematics in this article heavily exploits the CC sections feature. The CC feature assumes that the configuration space of a 3-D continuum architecture can be parameterized by three variables: arc length  $s$ , the curvature  $\kappa = 1/r$  as related to the radius  $r$  of a curve, and orientation  $\phi$  of the curve plane in space. The CC continuum bending can be decomposed into four following discrete motions:

- 1) a rotation to “point” the tangent at the curve’s origin to the curve’s end point;
- 2) a translation from curve origin to end;
- 3) a second rotation identical with the first to realign with the tangent at the curve’s end;
- 4) a rotation about the initial tangent (see Fig. 3).

**TABLE I**  
LINK PARAMETERS FOR VIRTUAL RRPR ROBOT MANIPULATOR

LINK	$\theta$	$d$	$a$	$\alpha$
1	$\theta_1$	0	0	90
2	$\theta_2$	0	0	-90
3	0	$d_3$	0	90
4	$\theta_4$	0	0	-90

Given this observation, a “virtual” 3-D four-joint rigid-link RRPR manipulator can be used to model the kinematic transformation along any CC backbone [33]. Consequently, the corresponding continuum robot forward kinematics model can be found using the conventional Denavit–Hartenberg [29] convention for the virtual robot (see Table I). The associated homogeneous transformation matrix of the virtual RRPR robot model is given as

$$H_4^0 = \begin{bmatrix} c_1 c_2 c_4 - c_1 s_2 s_4 & -s_1 & -c_1 c_2 s_4 - c_1 s_2 c_4 & -d_3 c_1 s_2 \\ s_1 c_2 c_4 - s_1 s_2 s_4 & c_1 & -s_1 c_2 s_4 - s_1 s_2 c_4 & -d_3 s_1 s_2 \\ s_2 c_4 + c_2 s_4 & 0 & -s_2 s_4 + c_2 c_4 & d_3 c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where  $s_l$  and  $c_l$  ( $l = 1, 2, 4$ ) denote  $\sin(\theta_l)$  and  $\cos(\theta_l)$  for the three revolute joints in the RRPR model, respectively, and  $d_3$  is the length of the third, prismatic, joint in the RRPR model. The continuum robot kinematics can be readily developed by substituting the joint variables of the virtual robot with the corresponding configuration space variables of a continuous curve. Specifically (see Fig. 3)

$$\theta_1 = \phi, \quad \theta_2 = \theta_4 = \theta, \quad \kappa = \left( \frac{1}{r} \right). \quad (2)$$

We have

$$s = r(2\theta) = \frac{(2\theta)}{\kappa} = \frac{(\theta_2 + \theta_4)}{\kappa} \rightarrow (\theta_2 + \theta_4) = s\kappa \quad (3)$$

$$\frac{d_3}{2} = r\sin\theta = \frac{\sin\theta}{\kappa} \rightarrow d_3 = \frac{2\sin\theta}{\kappa}. \quad (4)$$

Substituting (3) and (4) into the model (1) and simplifying gives

$$H_4^0 = \begin{bmatrix} c_\phi c_{s\kappa} & -s_\phi & -c_\phi s_{s\kappa} & (c_\phi - c_\phi c_{s\kappa})/\kappa \\ s_\phi c_{s\kappa} & c_\phi & -s_\phi s_{s\kappa} & (s_\phi c_{s\kappa} - s_\phi)/\kappa \\ s_{s\kappa} & 0 & c_{s\kappa} & s_{s\kappa}/\kappa \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

where  $c_m$  and  $s_m$  ( $m = s, \kappa, \phi, s\kappa$ ), denote  $\cos(m)$  and  $\sin(m)$ , respectively. The model (5) describes the forward kinematic relationship (3 by 3 orientation, top left of (5), and 3 by 1 translation, top right) between continuum curve shape/configuration space,  $C = [s \ \kappa \ \phi]^T$  (arc length  $s$ , curvature  $\kappa$ , and orientation  $\phi$ ), and task space,  $p = [x \ y \ z]^T$  ( $x$ ,  $y$ , and  $z$  are of the end-effector Cartesian coordinates).

2) *Virtual Robot Inverse Kinematics:* The inverse kinematics of the continuum robot can be represented by that of the spatial RRPR virtual robot. After the task space coordinates of the continuum robot are derived from the continuum robot forward

kinematics in (5), the  $x$ ,  $y$ , and  $z$  coordinates can, then, be substituted into the inverse kinematics of the RRPR robot model to obtain the desired joint space vector  $q = [\theta_1 \theta_2 d_3 \theta_4]^T$ . From (1), we obtain

$$x = -d_3 c_1 s_2, \quad y = -d_3 s_1 s_2, \quad z = d_3 c_2. \quad (6)$$

Since

$$x^2 + y^2 + z^2 = d_3^2 \left[ (c_1 s_2)^2 + (s_1 s_2)^2 + (c_2)^2 \right] = d_3^2. \quad (7)$$

Thus,

$$d_3 = \sqrt{x^2 + y^2 + z^2}. \quad (8)$$

The rotation  $\theta_2$  and  $\theta_4$  can be obtained from

$$\begin{aligned} \theta_2 &= \tan^{-1} \left( \frac{s_2}{c_2} \right) = \tan^{-1} \left( \frac{\sqrt{x^2 + y^2}}{d_3} / \frac{z}{d_3} \right) \\ &= \tan^{-1} \left( \frac{\sqrt{x^2 + y^2}}{z} \right) = \theta_4. \end{aligned} \quad (9)$$

The rotation  $\theta_1$  can also be derived as

$$\theta_1 = \tan^{-1} \left( \frac{s_1}{c_1} \right) = \tan^{-1} \left( \frac{y}{-d_3 s_2} / \frac{x}{-d_3 s_2} \right) = \tan^{-1} \left( \frac{y}{x} \right). \quad (10)$$

Analytical inverse kinematics is preferred over numerical inverse kinematics for its computational speed, which benefits real-time experimental implementation. Notice that the inverse kinematics in (8)–(10) has a unique solution as  $d_3$  is physically only positive and  $\theta_1$ ,  $\theta_2$ , and  $\theta_4$  have only one solution.

**3) Virtual Robot Dynamics:** Incorporating the dynamics of the continuum robot is vital for model-based dynamic control of continuum structures. The dynamic equations of motion, which provide the relationships between actuation and the acceleration, form the basis for several computational algorithms that are fundamental in control and simulation. In this article, the virtual RRPR rigid-link robot dynamics is derived and exploited to approximate the dynamics of a 3-D continuum architecture. For motion control, the dynamic model of a virtual RRPR mechanism is conveniently described by Lagrange dynamics represented in the joint-space formulation.

One continuum robot dynamic modeling approach is based on the Cosserat rod model, which is built upon a variable curvature framework. This model is resolved approximately by either numerical integration or by approximating the curvature function as a finite linear combination of basis functions [34]. However, the improvement in accuracy attained by the Cosserat rod model was not significant enough considering its computational and sensing cost [14]. Our proposed virtual robot model is built upon a CC assumption framework. The model represents continuum robot geometry with a well-known standardized approach that is modular and generally able to describe robots of any continuous shape. The virtual robot model is intuitive to understand, and it is sometimes easier to incorporate additional complex phenomena such as inertial dynamics. Two assumptions were made that further impact the design of the model and subsequent control

law. The approximated continuum robot dynamics is derived from virtual rigid-link dynamics, negating the elastic potential energy and damping effects. The continuum manipulator does not grasp objects, or otherwise contact the environment, to change its mass or dynamic properties.

Then, the Euler–Lagrange dynamics equations of the virtual RRPR manipulator can be written in a matrix form as

$$D(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) = \tau \quad (11)$$

where  $q = [\theta_1 \theta_2 d_3 \theta_4]^T \in \mathbb{R}^{4 \times 1}$  is the joint variable vector for the virtual RRPR robot. The vector  $\tau = [\tau_1 \tau_2 f_3 \tau_4]^T \in \mathbb{R}^{4 \times 1}$  is the torque applied to each joint variable. Specifically, the terms  $\tau_1$  is the applied torque at the first revolute joint, which drives the orientation of the continuum robot  $\tau_2$  and  $\tau_4$  are the applied torques at the second and fourth revolute joints, which form the shape curvature of the continuum robot, and  $f_3$  is the applied force at the third prismatic joint, which elongates/shrinks the continuum robot. When deriving the inertia matrix  $D(q) \in \mathbb{R}^{4 \times 4}$ , the third link inertia matrix ( $I_3$ ) is determined by assuming the third prismatic link of the virtual robot as a solid cylinder of uniform mass. The first, second, and fourth link inertia matrices ( $I_1$ ,  $I_2$ , and  $I_4$ ) are derived by modeling the links at the two ends of the prismatic joint as point masses. The detailed derivation process and derived terms of the inertia matrix  $D(q)$ , centrifugal and Coriolis matrix  $C(q, \dot{q}) \in \mathbb{R}^{4 \times 4}$ , and gravity matrix  $g(q) \in \mathbb{R}^{4 \times 1}$  can be found via the link <https://urlzs.com/wuEWb>.

## B. Two-Section Continuum Robot Model

The kinematics model for a two-section continuum robot inherits the one-section continuum robot kinematics model derived from Section II-A, with one additional definition. In order to preserve orientation during the forward kinematics calculation for multiple sections, a problem noted in [35], we define the local  $\phi$  of the distal section ( $\phi_2$ ) with respect to  $\phi$  of the proximal section ( $\phi_1$ ) as  $\phi_2 = \phi'_2 - \phi_1$ , where  $\phi'_2$  is the orientation of the distal section with respect to the base frame. The local task space coordinates of section 1 and 2,  $p_1 = [x_1 y_1 z_1]^T$  and  $p_2 = [x_2 y_2 z_2]^T$ , can be derived from configuration space variables of section 1,  $C_1 = [s_1 \kappa_1 \phi_1]^T$ , and section 2,  $C_2 = [s_2 \kappa_2 \phi_2]^T$ , using (5) to solve the forward kinematics. The inverse kinematics can be readily solved by deriving local joint space coordinates of section 1,  $q_1 = [\theta_1 \theta_2 d_3 \theta_4]^T$  and 2,  $q_2 = [\theta_5 \theta_6 d_7 \theta_8]^T$ , from  $p_1$  and  $p_2$  using (8)–(10). Consequently, the two-section continuum robot joint space vector  $q = [q_1^T q_2^T]^T \in \mathbb{R}^{8 \times 1}$ , task space vector  $p = [p_1^T p_2^T]^T \in \mathbb{R}^{6 \times 1}$ , and configuration space vector  $C = [C_1^T C_2^T]^T \in \mathbb{R}^{6 \times 1}$ , are obtained by concatenating variables from both continuum robot sections.

The dynamic model for a two-section continuum robot can be modeled by combining two virtual RRPR robots into an 8-DoF RRP RRPR rigid-link robot. The 8-DoF model of this virtual robot is constructed in a way so that the interface between the two RRPR robots aligns the last/first  $z$ -axes (see Fig. 3), as the 8-DoF virtual robot needs to model the continuum case, where the tangent between two sections are aligned. Sharing the same

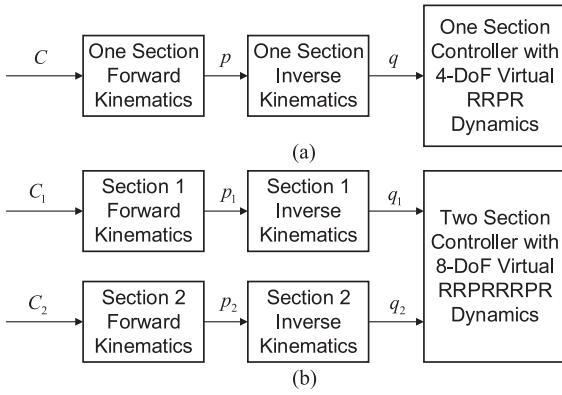


Fig. 4. (a) One-section and (b) two-section continuum robot model framework.

equation form as (11), the dynamics model for the virtual 8-DoF robot is derived via the Euler–Lagrange approach to compute torques  $\tau(q) \in \mathbb{R}^{8 \times 1}$ . For the sake of space and conciseness, the inertia matrix  $D(q) \in \mathbb{R}^{8 \times 8}$ , centrifugal and Coriolis matrix terms in  $C(q, \dot{q}) \in \mathbb{R}^{8 \times 8}$ , as well as the gravity matrix  $g(q) \in \mathbb{R}^{8 \times 1}$ , as derived by the authors for the 8-DoF virtual robot dynamics, is stored in the link <https://urlzs.com/vSmxy> for reference.

The procedure for mapping configuration space to joint space for the continuum robot dynamics and control is demonstrated in Fig. 4. The input desired trajectory was represented in the configuration space  $C$  and was used to calculate the desired continuum robot task space  $p$  using the forward kinematics in (5). Subsequently, the virtual robot variables  $q$  were derived from the inverse kinematics using (8)–(10) and fed into the control system as a desired reference input signal. Two input local coordinates ( $p = [p_1^T \ p_2^T]^T$ ), one for each section, are used for generating a unique joint space variable solution  $q = [q_1^T \ q_2^T]^T$ , preventing multiple solutions to inverse kinematics for the two-section model.

### III. CONTROL SYSTEMS DESIGN

The modeling strategies of the previous section form the basis for control approaches needed for continuum robots. We seek and exploit simple, relatively computationally inexpensive control methods used in (rigid-link) robot control systems [2] to design the controller in the virtual RRPR (one-section) and RRPRRRPR (two-section) robot coordinates. Multiple control methods, such as adaptive control [32], optimal and robust control [30], and learning control [36], are widely used in robotics. Each control method has advantages and disadvantages. However, the main aim of the system is to provide stability and high-frequency updates. In this article, we adopt the computed-torque [37] approach for the virtual robot, with the sensing and actuation transformed from and to the continuum robot, respectively.

The computed torque control consists of an inner nonlinear compensation loop and outer loop with an exogenous control

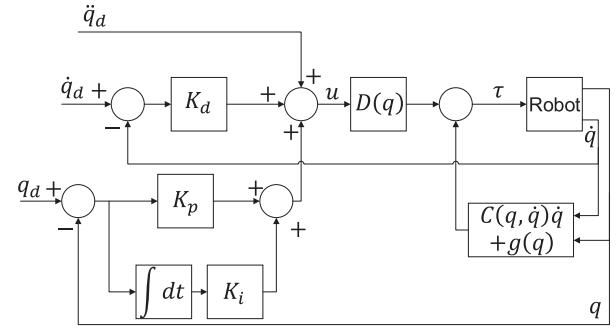


Fig. 5. Block diagram for computed-torque controller designed for virtual RRPR (one-section) and RRPRRRPR (two-section) robot.

signal  $u$ . The computed torque control law is given by

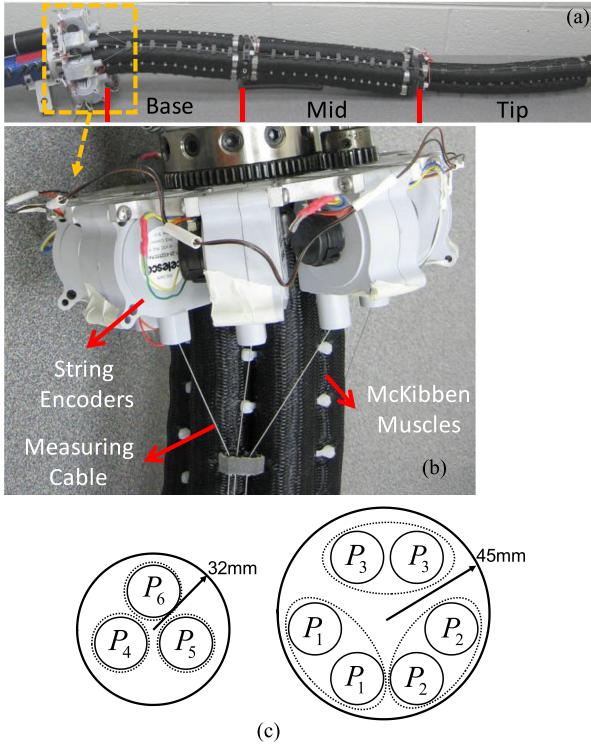
$$\begin{aligned} \tau &= D(q)[\ddot{q}_d + u]C(q, \dot{q})\dot{q} + g(q) \\ u &= K_d \dot{e} + K_p e + K_i \int e dt \end{aligned} \quad (12)$$

where  $K_d$ ,  $K_i$ , and  $K_p$  are symmetric positive definite design matrices and  $e = q_d - q$  denotes the position error. The term  $q_d$  expresses the desired virtual robot joint positions. Notice that the control law (12) contains the control signal  $u$ , which is of the proportional-integral-derivative (PID) type. However, the terms in  $u$  are premultiplied by the inertia matrix  $D(q_d - e)$ . Therefore, the computed torque control is not a linear controller as is the classic PID. The classic PID control can be inefficient because the controller contains ambiguity when environmental conditions or dynamics change. Experiments that compare the performance of proposed the dynamic control with that of PID control is presented in Section IV-D1.

The closed-loop equation is obtained by substituting the control action  $\tau$  from (12) in the robot dynamic model (11)

$$D(q)\ddot{q} = D(q)\left[\ddot{q}_d + K_d \dot{e} + K_p e + K_i \int e dt\right]. \quad (13)$$

The overall controller of the virtual robot is shown in Fig. 5. The values of the controller gains, i.e.,  $K_d$ ,  $K_i$ , and  $K_p$ , were determined according to an iterative experimental process to maximize controller performance. The desired joint space vector,  $q_d \in \mathbb{R}^{n \times 1}$  ( $n = 4$  for one-section and  $n = 8$  for two-section), derived from the desired configuration space in Section II, served as the control system setpoint. Their first- and second-order derivatives were numerically estimated and served as the inputs to the controller through  $\mathbb{R}^{n \times 1}$  vectors  $\dot{q}_d$ ,  $\ddot{q}_d$ , and  $\ddot{\dot{q}}_d$ . The output of the controller  $u \in \mathbb{R}^{n \times 1}$  was then used to establish the torque signal  $\tau \in \mathbb{R}^{n \times 1}$  along with systems  $D(q) \in \mathbb{R}^{n \times n}$ ,  $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ , and  $g(q) \in \mathbb{R}^{n \times 1}$  matrices. Next, the control action  $\tau \in \mathbb{R}^{n \times 1}$  was converted to actuation space in the form of pressure and applied to the physical continuum robot, which in turn feeds back the current continuum robot shape configuration via encoder data. The data were subsequently converted to virtual robot rotation and translation signal vector  $q \in \mathbb{R}^{n \times 1}$ , whose first derivative  $\dot{q}$  was numerically estimated and input to the controller to form the error and drive the control action.



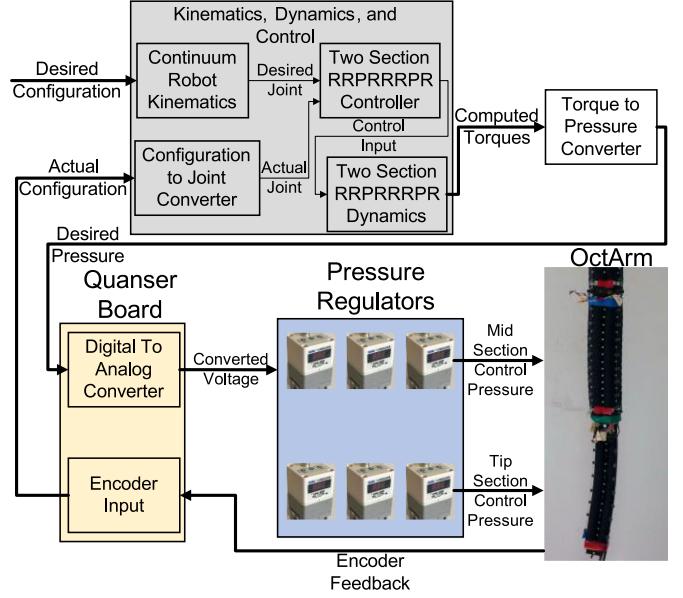
**Fig. 6.** (a) OctArm manipulator with base, mid, and tip sections. (b) OctArm-based-mounted draw wire strain sensors. (c) OctArm actuator cross-section configurations for tip (left) and mid (right) sections (Dotted lines show three control channels).

#### IV. EXPERIMENTAL IMPLEMENTATION

##### A. Experimental Setup

To further demonstrate the validity of the proposed controller based on the virtual rigid-link dynamics model, the experiments are conducted on the tip and mid-sections of the OctArm continuum manipulator [27], [28]. The OctArm, whose structure is shown in Fig. 6(a), is a nine-DoF pneumatically actuated, extensible, continuum robot capable of motion in three dimensions. The kinematically redundant manipulator is comprised of three serially connected sections: base, mid, and tip-section. Each of the three sections can extend (with arc length  $s$ ) and bend in any direction (with curvature  $\kappa$  and orientation  $\phi$ ), providing three DoF for each section. The OctArm is constructed using compressed air-actuated McKibben extension muscles with three control channels per section (see Fig. 6(c)) [27]. The mid-section has three pairs of McKibben actuators spaced radially at 120° intervals, forming one control channel per pair. Such mid-section design leads the muscle actuators to be designed for a larger radius and results in higher stiffness and load capacity, though at the expense of manipulability. The tip-section of the device is comprised of three single McKibben actuators arranged radially at 120° intervals, resulting in greater manipulability as compared to the mid-section.

To acquire the configuration space (arc length  $s$ , curvature  $\kappa$ , and orientation  $\phi$ ) of the OctArm in real-time, shape sensing is added to the robot by mounting nine draw wire encoders at the base section: one for each control channel [see Fig. 6(b)].



**Fig. 7.** Experimental control loop block diagram for two section OctArm virtual eight-DoF RRPBRRPR discrete-jointed model dynamics control.

Each encoder senses the length of the respective control channel, and the shape of each manipulator section is inferred assuming constant section curvature.

A set of experiments utilizing the OctArm and the described model were implemented. The model and controller were implemented in MATLAB/Simulink environment [38]. Interfacing with the OctArm was accomplished using two Quanser Q8-USB data acquisition boards [39]. State estimation of the system was provided through internal measurements of the OctArm via a series of string encoders that run along the length of each section muscle. After output torques and forces are computed from the proposed controller, they are converted to pneumatic pressures via a series of pressure regulators using an output voltage from the Quanser boards. The pressures are, then, applied onto the corresponding McKibben extension muscles, where one regulator is assigned to one muscle or one muscle pair in the case of the mid-section (see Fig. 7). A full torque/force to pneumatic pressure conversion technique is formulated and discussed in Section V-B. Experiments on both one section (tip-section, 3-DoF) and two sections (tip and mid-sections, 6-DoF) are conducted in 3-D space to examine configuration space tracking performance of the proposed control method for multisegment continuum robots. Results of experiments are presented in Sections V-C and V-D and, with discussions and future research contained in Section V-E.

##### B. Torque to Pressure Conversion

To facilitate the implementation of the proposed controller into pneumatically actuated continuum architectures like the OctArm, the computed torques  $[\tau_1 \tau_2 f_3 \tau_4]^T$  for controlling the virtual rigid-link robot model in (16) must be converted into applied pneumatic pressures onto each of the pneumatic "McKibben" muscles in the single tip-section. Such conversions

can be inspired by the movements of the OctArm, which can be categorized into three distinct motions: 1) extension/contraction, which determines the continuum robot arc length  $s$ ; 2) bending, which accounts for the OctArm curvature  $\kappa$ ; and (3) torsion, which translates to continuum robot orientation  $\phi$ .

For pure extension/contraction motion, the calculated extension force  $f_3$  that results from the RRPR model is equally applied to the three muscles to achieve balanced pure extending movement. In order to generate simultaneous extending, bending, and torsion motions at the single tip-section of the OctArm shown in Fig. 6(c) left, the controller generated torques  $\tau_1$ ,  $\tau_2$ , and  $\tau_4$ , which are responsible for driving the rotation  $\phi$  and curvature  $\kappa$  need to be incorporated. Therefore, the pressure applied to the three tip-section McKibben muscle control channels can be represented as

$$P_4 = k_p [f_3 + \Delta\tau_t \cdot \cos(\phi_t + b \cdot \tau_1)] \quad (14a)$$

$$P_5 = k_p [f_3 + \Delta\tau_t \cdot \cos(\phi_t + b \cdot \tau_1 + 120^\circ)] \quad (14b)$$

$$P_3 = k_p [f_3 + \Delta\tau_t \cdot \cos(\phi_t + b \cdot \tau_1 - 120^\circ)] \quad (14c)$$

where  $\Delta\tau_t = (\tau_2 + \tau_4)/2$ ,  $\phi_t$  denotes the current tip-section rotation,  $k_p$  is the conversion gain from torque to pressure, and  $b$  is a constant. In (14), the terms  $f_3$ ,  $\Delta\tau_t$ , and  $\tau_1$  account for extending, bending, and torsion maneuvers of the tip-section, respectively. The difference of pressure given to three distinct sets of control channels will generate a bending effect of constant curvature that matches the continuum robot kinematics model.

Pressure conversion of the two-section OctArm, which can be modeled as a virtual RRPRRRPR robot is achieved similarly to that of the one section. After the derivation of the torque vector  $\tau = [\tau_1 \ \tau_2 \ f_3 \ \tau_4 \ \tau_5 \ \tau_6 \ f_7 \ \tau_8]^T$ , whose first four terms model the mid-section and last four terms model the tip-section, the pressures on three mid-section control channels  $P_1$ ,  $P_2$ , and  $P_3$  and the pressures on three tip-section control channels  $P_4$ ,  $P_5$ , and  $P_6$  [shown in Fig. 6(c)], are determined by

$$P_1 = k_p [f_3 + \Delta\tau_m \cdot \cos(\phi_m + b \cdot \tau_1)] \quad (15a)$$

$$P_2 = k_p [f_3 + \Delta\tau_m \cdot \cos(\phi_m + b \cdot \tau_1 + 120^\circ)] \quad (15b)$$

$$P_3 = k_p [f_3 + \Delta\tau_m \cdot \cos(\phi_m + b \cdot \tau_1 - 120^\circ)] \quad (15c)$$

$$P_4 = k_p [f_7 + \Delta\tau_t \cdot \cos(\phi_t + c \cdot \tau_5)] \quad (15d)$$

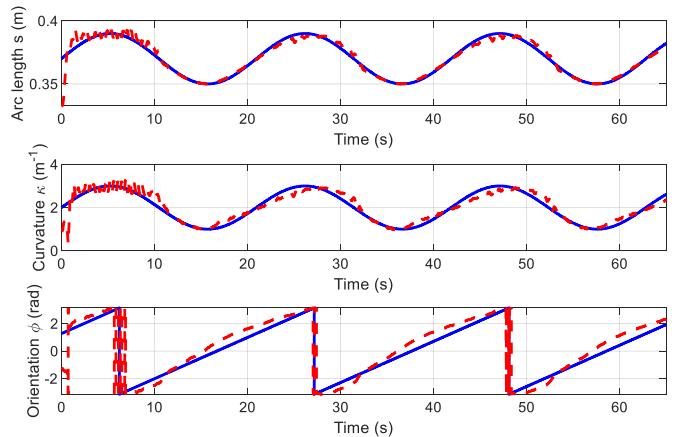
$$P_5 = k_p [f_7 + \Delta\tau_t \cdot \cos(\phi_t + c \cdot \tau_5 + 120^\circ)] \quad (15e)$$

$$P_6 = k_p [f_7 + \Delta\tau_t \cdot \cos(\phi_t + c \cdot \tau_5 - 120^\circ)] \quad (15f)$$

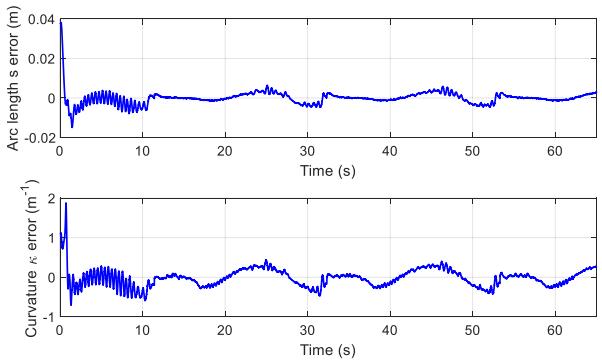
where  $\Delta\tau_m = (\tau_2 + \tau_4)/2$  drives the mid-section bending behavior,  $\Delta\tau_t = (\tau_6 + \tau_8)/2$  drives the tip-section bending,  $\phi_m$  denotes the current mid-section rotation, and  $c$  is a constant. The values of  $k_p$ ,  $b$ , and  $c$  are first approximated to ensure that the converted pressure falls in the reasonable range in which the OctArm can operate properly. Then, the values are finely tuned through iterative experimental process to maximize controller performance.

### C. One-Section OctArm Experiment

The first experiment conducted on the OctArm manipulator reported here is simultaneous extension/contraction, bending,

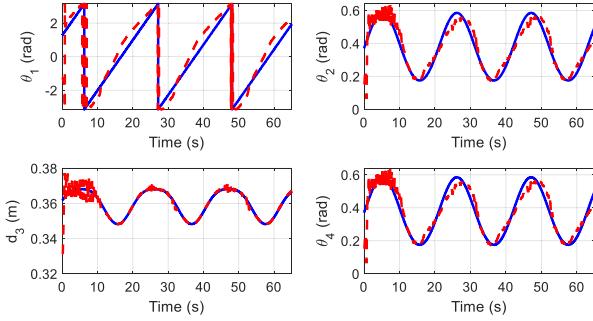


**Fig. 8.** One section experiment (OctArm tip section) — Desired (blue solid line) and actual (red dashed line) arc length  $s$ , curvature  $\kappa$ , and orientation  $\phi$  in configuration space.



**Fig. 9.** One section experiment (OctArm tip section) — Arc length  $s$ , and curvature  $\kappa$  error on OctArm in configuration space.

and torsion on one OctArm section. The experiment includes the one-section continuum robot model derived in Section II-A and intends to actuate only the Octarm tip-section. Detailed video footage of the experiment can be found via the link <https://urlzs.com/HiYMH>. In this experiment, the system is fed desired arc length  $s$  a sinusoid with an amplitude 0.03 m and a frequency of 0.08 Hz, as shown in Fig. 8. The desired curvature  $\kappa$  is also a sinusoidal signal with an amplitude of  $1\text{m}^{-1}$  and a frequency of 0.08 Hz while the desired orientation  $\phi$  is a ramp function with a slope of 0.3 (rad/s) to generate a full  $360^\circ$  rotation on the tip-section. During the experiment, the OctArm initiated from its natural unpressurized length of 0.34 m and rapidly converged to the desired arc length with minor error in the crest of the sine wave. The actual arc length  $s$  and curvature  $\kappa$  settle relatively fast and no obvious overshoot or oscillations are detected. Possibly due to intrinsic settling nature of the PID controller, the high-frequency oscillatory motion observed in the first cycle of arc length and curvature sinusoid terminates after 7 s before the system reaches a steady state. Also, the robot accomplishes three full  $360^\circ$  rotations in conjunction with the sinusoidal arc length and curvature variations. Highlighting the effectiveness of the controller, the arc length, and curvature error plot illustrated in Fig. 9 show that the control algorithm



**Fig. 10.** One section experiment (OctArm tip section) — Desired and actual  $\theta_1$ ,  $\theta_2$ ,  $d_3$ , and  $\theta_4$  of the virtual RRPR rigid-link model in OctArm joint space.

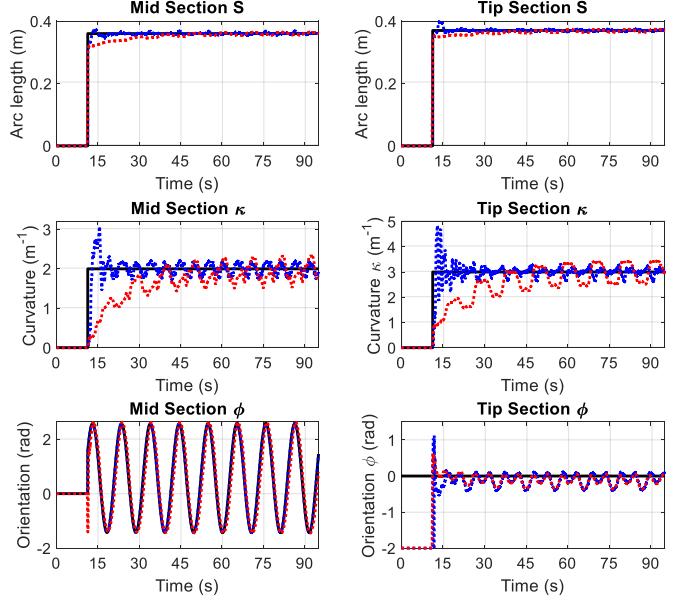
implemented on the OctArm only outputs arc length and curvature error of  $\pm 5\text{mm}$  and  $\pm 0.35 \text{ m}^{-1}$ , respectively, which is considered within a reasonable range for such a complex maneuver with this robot.

To further demonstrate the control performance of the one section continuum robot during concurrent extending, bending, and rotation in joint space, the virtual RRPR joint variables of the OctArm tip-section are displayed in Fig. 10. Directly derived from the encoder generated OctArm configuration space variables, the joint space variables not only demonstrate the efficacy of the controller but also reflect the accuracy of the proposed continuum architecture approximation using virtual RRPR rigid-link model. The actual joint variable values from the OctArm encoders track the desired joint variables from the virtual RRPR robot inverse kinematics relatively well. The oscillation at the beginning of the experiment, and minor error at the sinusoid crest observed in the configuration space occurred in the joint space as well.

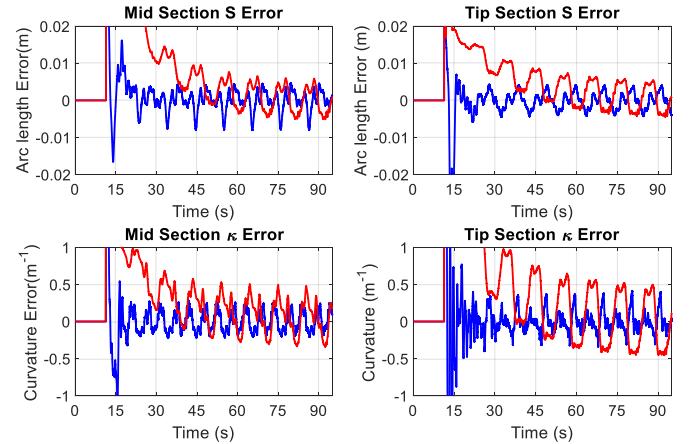
#### D. Two-Section OctArm Experiment

The models for the complete, two-section continuum robot are formed in the Section II-B by deriving the forward kinematics, inverse kinematics, and dynamics for the virtual RRP<sub>RRR</sub> rigid-link robot. After implementing the models into the system described in Fig. 2, the computed torques applied on all eight virtual joints are converted to six pressure signals, which are used to actuate the OctArm mid and tip-sections. In this section, three experiments were conducted to study (1) the difference between classic static PID controller versus the proposed dynamic controller, (2) the controller's ability to handle fast maneuvers, and (3) the controller's performance under complex manipulator trajectory.

**1) Two Section Experiment 1: Classic PID versus Our Approach:** In a first experiment, the proposed dynamic model-based controller is tested against the classic static PID controller. Therefore, the dynamic terms,  $D(q)$ ,  $C(q, \dot{q})$ , and  $g(q)$ , along with the feedforward term  $\ddot{q}_d$  in (12) are deactivated, such that only PID terms are active ( $\tau = K_p e + K_d \dot{e} + K_i \int e dt$ ). The gains  $K_p$ ,  $K_d$ , and  $K_i$  used for the static PID and proposed dynamic controller are identical. The desired arc length and curvature for both sections remains constant:  $s_{\text{mid}} = 0.36 \text{ m}$ ,



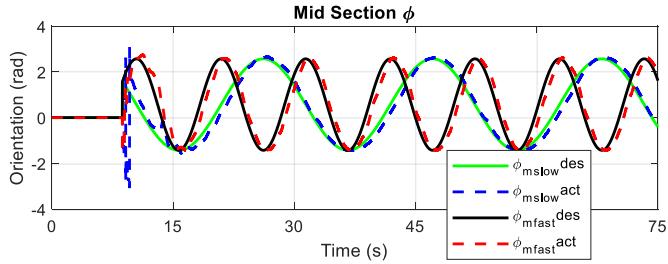
**Fig. 11.** Two section experiment 1 (OctArm tip and mid sections) — Desired (black solid line) and actual arc length  $s$ , curvature  $\kappa$ , and orientation  $\phi$  in configuration space (dynamic controller in blue dotted line, and classic PID in red dotted line).



**Fig. 12.** Two section experiment 1 (OctArm tip and midsection) — Arc length  $s$ , and curvature  $\kappa$  error on OctArm in configuration space (dynamic controller in blue line, and classic PID in red line).

$s_{\text{tip}} = 0.37 \text{ m}$ ,  $\kappa_{\text{mid}} = 2 \text{ m}^{-1}$ , and  $\kappa_{\text{tip}} = 3 \text{ m}^{-1}$ . The mid-section follows a sinusoidal rotation  $\phi_{\text{mid}} = 2 \sin(0.6t) - 1$  whereas the tip section has no desired torsional motion.

The desired and actual arc configuration space variables for both dynamic model-based controller and the classic static PID controller are presented in Fig. 11. Both controllers achieved decent arc length tracking performance:  $\pm 4 \text{ mm}$  error for dynamic controller and  $\pm 5 \text{ mm}$  error for PID controller during steady state (see Fig. 12). However, the dynamic controller rose to and settled at the steady state much faster than the PID controller. A similar trend was observed in the mid- and tip-section curvature tracking, which further verified excellent performance delivered by the dynamic controller. Not only was the dynamic controller's error much smaller (see Fig. 13),



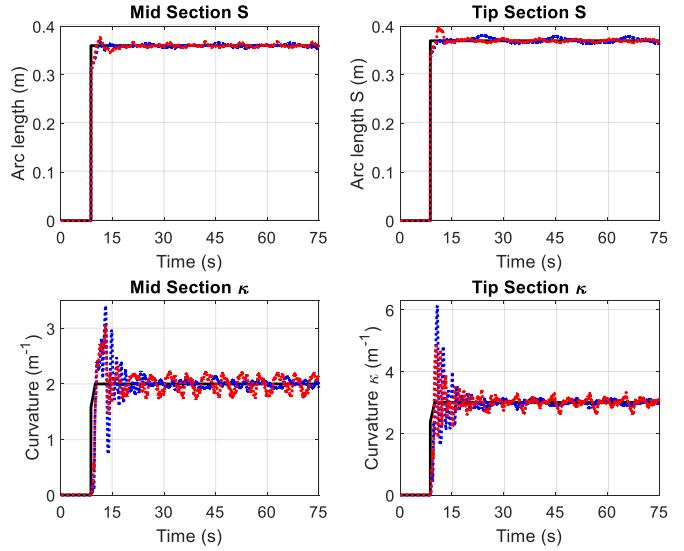
**Fig. 13.** Two section experiment 2 (OctArm tip and mid sections) — Desired and actual mid-section orientation under slow and fast speed rotations.

but also its rise and settling times were much shorter, a fact that ensures superior responsiveness at the sacrifice of minor overshoot. The mid and tip curvature error improvement of the dynamic controller compared to the PID controller is 48.1% and 20.5%, respectively. We improved the rise time by 94.8%, the dynamic controller's rise time for mid-section curvature is 0.92 s while the PID controller's rise time is 17.62 s. The proposed virtual RRPURRPR dynamics model based dynamic controller captures the nonlinearities of continuum robots and is important for industrial applications where time is important along with the accuracy.

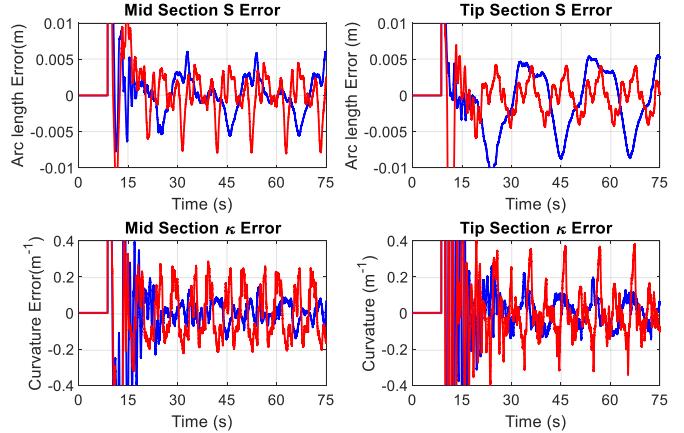
**2) Two Section Experiment 2: Slow Rotation versus Fast Rotation:** The first two-section experiment features two tests with the same desired mid and tip sections curvature and arc length. However, mid-section rotation of the second test is two times faster than that of the first test:  $\phi_{mslow}^{des} = 2 \sin(0.3t) - 1$  and  $\phi_{mfast}^{des} = 2 \sin(0.6t) - 1$  (see Fig. 13). Detailed video footage of the both tests can be found via the link <https://shorturl.at/kzFU8> (slow) and <https://shorturl.at/nyY28> (fast). This experiment is focused on trajectory tracking performance distinction between the two tests with the aim to examine the reliability of the 8-DoF dynamics and control framework under high-speed working conditions.

The desired and actual configuration space variables for the first experiment are shown in Fig. 14 and the configuration space tracking error can be found in Fig. 15. Both tests achieved remarkable performance regarding mid and tip section arc length tracking. The slower experiment demonstrated slightly worse arc length trajectory following ability but better curvature tracking performance than the faster one, as shown in Fig. 15. In addition, the robot with faster mid-section rotation demonstrated smaller percent overshoot and shorter settling time during curvature tracking than the robot with slower mid-section rotation. The sinusoidal torsional displacement of the mid-section is accomplished smoothly by both tests. Collectively, the 8-DoF dynamics model-based controller demonstrates excellent performance with decent error convergence and fast response in the configuration space for both slow and fast maneuvers.

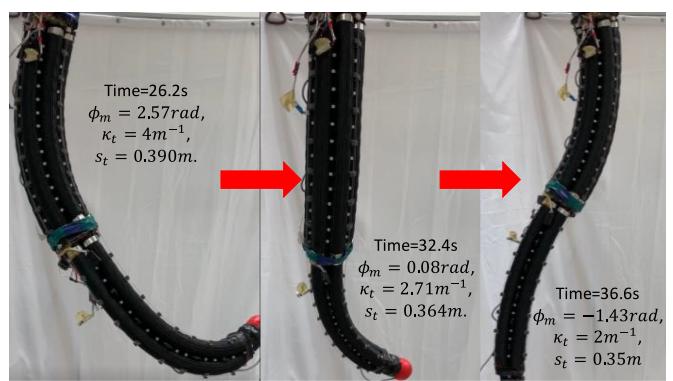
**3) Two Section Experiment 3: More Demanding Maneuver:** A more complicated maneuver with a sinusoidal tip-section curvature and arc length trajectory (see Fig. 16), whose video footage link is <https://shorturl.at/iszP2>, is imposed to examine the control method resilience against demanding continuum robot operations. The desired and actual configuration space



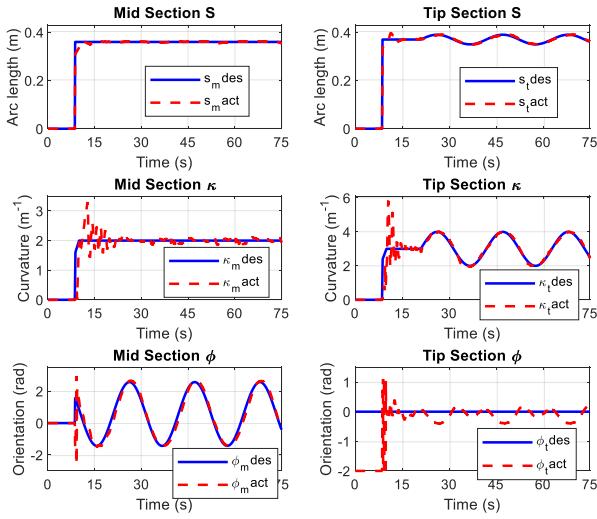
**Fig. 14.** Two section experiment 2 (OctArm tip and mid sections) — Desired (black solid line) and actual arc length  $s$  and curvature  $\kappa$  in configuration space (slow rotation in blue dotted line, and fast rotation in red dotted line).



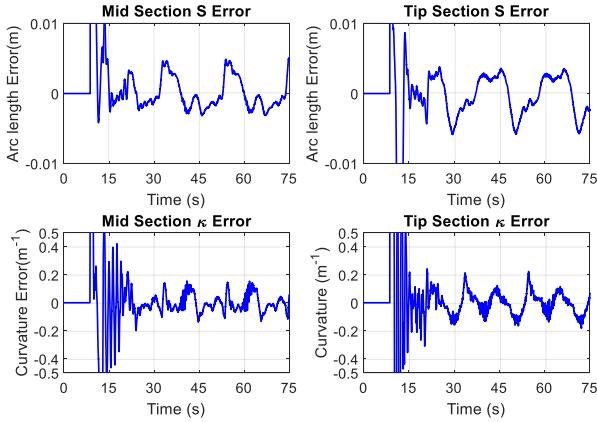
**Fig. 15.** Two section experiment 2 (OctArm tip and midsection) — Arc length  $s$ , and curvature  $\kappa$  error on OctArm in configuration space (slow rotation in blue line, and fast rotation in red line).



**Fig. 16.** Two section experiment 3: Mid-section rotates from 2.57 rad to -1.43 rad while the tip section curvature and arc length follows a sinusoidal path. The tip section at 26.2s is more curled than the tip section at 36.6s.



**Fig. 17.** Two section experiment 3 (OctArm tip and mid sections)—Desired and actual arc length  $s$ , curvature  $\kappa$ , and orientation  $\phi$  in configuration space.



**Fig. 18.** Two section experiment 3 (OctArm tip and mid sections) — Arc length  $s$ , and curvature  $\kappa$  error in configuration space.

variables are exhibited in Fig. 17. The arc length for both tip and mid-section display outstanding performance with minor overshoot on the tip-section. Although the mid-section follows the desired curvature trajectory well, it exhibits relatively large overshoot and long settling time of approximately 9.3 s. The tip-section curvature also experiences a major overshoot before it reaches a steady state after 15 s. Both the tip and mid-section track the desired orientation well. Collectively, experiment 3 with a more demanding tip-section trajectory to follow maintained a decent tracking performance, as shown in Fig. 18, where configuration variable errors for both sections remained in a reasonable interval.

### E. Discussion and Future Research

The main objective of the experiments was to evaluate the tracking capability of the proposed controller for multisection continuum robots in configuration space. There are four critical findings inferred from the experimental evaluations on the

OctArm. First, the virtual discrete-jointed robot model-based controller accomplishes the configuration space tracking for both the single-section and two-section OctArm trajectories satisfactorily. Second, the proposed dynamic model-based control, which output less error and faster rise and settling time, proves to be superior to the classic PID control. The dynamic control also has approximately 75% less arc length error and less chattering than the SMC implemented by Kapadia [30]. The chattering from the SMC causes high wear of moving continuum robot mechanical parts and is an obstacle for its implementation. Third, the 8-DoF dynamics based two-section continuum robot control design is capable of tracking high frequency signals and following complicated configuration space trajectories. Finally, the configuration space errors exhibit sequential and cyclical nature in both one-section and two-section experiments. The cyclical pattern observed is likely due to unmodeled dynamics and the discrepancies between the physical system and the ideal model used. Corrections made to the model as well as compensation to the physical system could reduce or eliminate these behaviors.

Further studies should investigate the disturbance rejection capabilities of the proposed control method. Also, potential energy terms for elastic and damping effects can be added to the proposed virtual discrete-jointed dynamics model to capture the oscillatory and elastic nature of continuum robots and pneumatic actuators [40]. Finally, upon the successful realization of configuration space control, the proposed dynamic control framework should be examined for its task-space tracking capability.

## V. CONCLUSION

In this article, a novel model-based dynamic feedback control architecture was introduced for spatial multisection continuum robots. Inspired by conventional rigid link robot computed-torque control techniques, the control law utilizes the model of a virtual, conventional rigid link robot with discrete joints, in whose coordinates the controller is developed, to generate real-time control inputs for the continuum robot. The computed-torque input was translated to pneumatic pressures applied to each pneumatic artificial muscle in the actuator space of the continuum robot through a carefully designed converter. This controller was implemented on the OctArm—a pneumatically actuated spatial continuum manipulator with three sections—using a control architecture with both feedback linearization and PID controller. The forward and inverse kinematics, as well as dynamics model approximated by a virtual discrete-jointed robot model, are derived for both single and two-section continuum robots. The proposed controller was experimentally validated on both single and two sections of the OctArm. Results from the test demonstrate that the curvature tracking error and rise time can be reduced up to 48.1% and 94.8% with the proposed dynamic controller when compared to the traditional PID controller during multisection maneuvers. Accompanying this article is video footage showing the configuration space tracking motion of the OctArm in the 3-D space as reported in the experimental results.

## REFERENCES

- [1] G. Robinson and J. B. C. Davies, "Continuum robots - a state of the art," in *Proc. IEEE Int. Conf. Robot. Automat.*, Detroit, MI, USA, 1999, vol. 4, pp. 2849–2854.
- [2] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1661–1683, Nov. 2010.
- [3] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Appl. Bionics Biomechanics*, vol. 5, no. 3, pp. 99–117, Dec. 2008.
- [4] I. D. Walker, "Continuous backbone "Continuum" robot manipulators," *ISRN Robot.*, vol. 2013, pp. 1–19, 2013.
- [5] D. B. Camarillo, C. R. Carlson, and J. K. Salisbury, "Configuration tracking for continuum manipulators with coupled tendon drive," *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 798–808, Aug. 2009.
- [6] D. B. Camarillo, C. R. Carlson, and J. K. Salisbury, "Task-space control of continuum manipulators with coupled tendon drive," in *Experimental Robotics*, O. Khatib, V. Kumar, and G. J. Pappas, Eds., vol. 54. Berlin, Germany: Springer, 2009, pp. 271–280.
- [7] A. Bajo, R. E. Goldman, and N. Samaan, "Configuration and joint feedback for enhanced performance of multi-segment continuum robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 2905–2912.
- [8] A. Bajo and N. Samaan, "Hybrid motion/force control of multi-backbone continuum robots," *Int. J. Robot. Res.*, vol. 35, no. 4, pp. 422–434, Apr. 2016.
- [9] M. Mahvash and P. E. Dupont, "Stiffness control of surgical continuum manipulators," *IEEE Trans. Robot.*, vol. 27, no. 2, pp. 334–345, Apr. 2011.
- [10] T. Mahl, A. Hildebrandt, and O. Sawodny, "A variable curvature continuum kinematics for kinematic control of the bionic handling assistant," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 935–949, Aug. 2014.
- [11] H. Wang, W. Chen, X. Yu, T. Deng, X. Wang, and R. Pfeifer, "Visual servo control of cable-driven soft robotic manipulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, Nov. 2013, pp. 57–62.
- [12] B. Conrad and M. Zinn, "Closed loop task space control of an interleaved continuum-rigid manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Seattle, WA, USA, May 2015, pp. 1743–1750.
- [13] A. D. Marchese and D. Rus, "Design, kinematics, and control of a soft spatial fluidic elastomer manipulator," *Int. J. Robot. Res.*, vol. 35, no. 7, pp. 840–869, Jun. 2016.
- [14] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft Robot.*, vol. 5, no. 2, pp. 149–163, Apr. 2018.
- [15] I. A. Gravagne and I. D. Walker, "Uniform regulation of a multi-section continuum manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, DC, USA, 2002, vol. 2, pp. 1519–1524.
- [16] V. Falkenhahn, A. Hildebrandt, and O. Sawodny, "Trajectory optimization of pneumatically actuated, redundant continuum manipulators," in *Proc. Amer. Control Conf.*, Portland, OR, USA, Jun. 2014, pp. 4008–4013.
- [17] A. D. Marchese, R. Tedrake, and D. Rus, "Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Seattle, WA, USA, May 2015, pp. 2528–2535.
- [18] V. Falkenhahn, A. Hildebrandt, R. Neumann, and O. Sawodny, "Dynamic control of the bionic handling assistant," *IEEE/ASME Trans. Mechatron.*, vol. 22, no. 1, pp. 6–17, Feb. 2017.
- [19] C. Wang, J. Wagner, C. G. Frazelle, and I. D. Walker, "Continuum robot control based on virtual discrete-jointed robot models," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc.*, USA, Oct. 2018, pp. 2508–2515.
- [20] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *J. Robot. Syst.*, vol. 20, no. 2, pp. 45–63, Feb. 2003.
- [21] Z. Tang, H. L. Heung, K. Y. Tong, and Z. Li, "A novel iterative learning model predictive control method for soft bending actuators," in *Proc. IEEE Int. Conf. Robot. Autom.*, Montreal, Canada, May 2019, pp. 4004–4010.
- [22] R. K. Katschmann, C. D. Santina, Y. Toshimitsu, A. Bicchi, and D. Rus, "Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model," in *Proc. 2nd IEEE Int. Conf. Soft Robot.*, Seoul, South Korea, Apr. 2019, pp. 454–461.
- [23] T. Greigarn, N. L. Poiriot, X. Xu, and M. C. Cavusoglu, "Jacobian-based task-space motion planning for MRI-actuated continuum robots," *IEEE Robot. Autom. Lett.*, vol. 4, no. 1, pp. 145–152, Jan. 2019.
- [24] W. S. Rone and P. Ben-Tzvi, "Continuum robot dynamics utilizing the principle of virtual power," *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 275–287, Feb. 2014.
- [25] I. S. Godage, G. A. Medrano-Cerda, D. T. Branson, E. Guglielmino, and D. G. Caldwell, "Dynamics for variable length multisection continuum arms," *Int. J. Robot. Res.*, vol. 35, no. 6, pp. 695–722, May 2016.
- [26] J. Till, V. Alois, and C. Rucker, "Real-time dynamics of soft and continuum robots based on Cosserat rod models," *Int. J. Robot. Res.*, vol. 38, no. 6, pp. 723–746, May 2019.
- [27] M. D. Grissom et al., "Design and experimental testing of the OctArm soft robot manipulator," in *Proc. SPIE*, vol. 6230, no. 62301F, Orlando, Florida, USA, May 2006.
- [28] I. D. Walker et al., "Continuum robot arms inspired by cephalopods," in *Proc. SPIE*, vol. 5804, Orlando, Florida, USA, May 2005, pp. 303–314.
- [29] A. Kapadia and I. D. Walker, "Task-space control of extensible continuum manipulators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, Sep. 2011, pp. 1087–1092.
- [30] A. D. Kapadia, I. D. Walker, D. M. Dawson, and E. Tatliciooglu, "A model-based sliding mode controller for extensible continuum robots," in *Proc. 9th Int. Conf. Signal Process., Robot. Automat.*, Stevens Point, Wisconsin, USA, 2010, pp. 113–120.
- [31] A. D. Kapadia, K. E. Fry, and I. D. Walker, "Empirical investigation of closed-loop control of extensible continuum manipulators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Chicago, IL, USA, Sep. 2014, pp. 329–335.
- [32] C. G. Frazelle, A. D. Kapadia, and I. D. Walker, "A nonlinear control strategy for extensible continuum robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, Brisbane, QLD, Australia, May 2018, pp. 7727–7734.
- [33] M. W. Hannan and I. D. Walker, "Analysis and experiments with an elephant's trunk robot," *Adv. Robot.*, vol. 15, no. 8, pp. 847–858, Jan. 2001.
- [34] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1261–1280, Dec. 2015.
- [35] B. A. Jones and I. D. Walker, "A new approach to jacobian formulation for a class of multi-section continuum robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, Barcelona, Spain, 2005, pp. 3268–3273.
- [36] D. Braganza, D. M. Dawson, I. D. Walker, and N. Nath, "A neural network controller for continuum robots," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1270–1277, Dec. 2007.
- [37] R. Middlestone and G. Goodwin, "Adaptive computed torque control for rigid link manipulators," in *Proc. 25th IEEE Conf. Decis. Control*, Athens, Greece, Dec. 1986, pp. 68–73.
- [38] "Simulink - Simulation and Model-Based Design." [Online]. Available: <https://www.mathworks.com/products/simulink.html>, Accessed: Jun. 18, 2019.
- [39] "Q8-USB Data Acquisition Device - Quanser." [Online]. Available: <https://www.quanser.com/products/q8-usb-data-acquisition-device/>, Accessed: Jun. 18, 2019.
- [40] E. Tatliciooglu, I. D. Walker, and D. M. Dawson, "New dynamic models for planar extensible continuum robot manipulators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, CA, USA, Oct. 2007, pp. 1485–1490.



**Chengshi Wang** received the B.S. degree in mechanical engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2014 and the M.S. degree in mechanical engineering from Clemson University, Clemson, SC, USA, in 2019, where he is currently working toward the Ph.D. degree in nonlinear modeling and control of driving interfaces and continuum/soft robots for system performance gains with the Department of Mechanical Engineering.

His current research interests include dynamical systems and control, robotics and mechatronics, and haptics for ground vehicles.



**Chase G. Frazelle** received the B.Sc. and M.S. degrees in electrical engineering, in 2015 and 2017, respectively, from Clemson University, Clemson, SC, USA, where he is currently working toward the Ph.D. degree in electrical engineering.

His research interests include the control and modeling of extensible continuum robots with the aim to improve the reliability and automation of such manipulators for space exploration and investigation, human-robot interfacing, embedded systems, and interaction with distributed systems.



**John R. Wagner** (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in mechanical engineering from the State University of New York at Buffalo, Buffalo, NY, USA and from Purdue University, West Lafayette, IN, USA, in 1983, 1985, and 1989, respectively. He is currently the Director of the Product Lifecycle Management Center, Clemson University. His research interests include nonlinear control theory, smart cooling systems, diagnostic/prognostic strategies, and mechatronic system design with application to transportation and power generation systems.

Dr. Wagner is a Licensed Professional Engineer and a Fellow of both the American Society of Mechanical Engineers and Society of Automotive Engineers.



**Ian D. Walker** (Fellow, IEEE) received the B.Sc. degree in mathematics from the University of Hull, Hull, U.K., in 1983, and the M.S. and Ph.D. degrees in electrical engineering from the University of Texas at Austin, Austin, TX, USA, in 1985 and 1989, respectively.

His research interests include biologically inspired and continuum robotics, as well as architectural robotics.

Dr. Walker was a Vice President for Financial Activities for the IEEE Robotics and Automation Society, and as a Chair of the AIAA Technical Committee on Space Automation and Robotics. He was on the Editorial Boards of the IEEE TRANSACTIONS ON ROBOTICS, the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, the *International Journal of Robotics and Automation*, the IEEE ROBOTICS AND AUTOMATION MAGAZINE, and the *International Journal of Environmentally Conscious Design and Manufacturing*. He is currently with the Editorial Board of Soft Robotics.