Continuum Robot Control Based on Virtual Discrete-Jointed Robot Models

Chengshi Wang and John Wagner
Department of Mechanical Engineering
Clemson University
Clemson, SC 29634, USA
{chengsw & jwagner}@clemson.edu

Chase G. Frazelle and Ian D. Walker
Department of Electrical and Computer Engineering
Clemson University
Clemson, SC 29634 USA
{cfrazel & iwalker}@clemson.edu

***Abstract*—Continuum (continuous backbone) robots are suitable for operation in unstructured environments thanks to their inherent compliance. They can adjust their shape to navigate through complex environments and grasp a wide variety of payloads with their compliant backbones. However, controller design for continuum robots is challenging due to their complex dynamics. In this paper, we introduce a new and novel strategy for trajectory control of continuum robot sections. The approach is based on a virtual discrete-jointed robot whose degrees of freedom are directly mapped to those of a continuum robot section. A conventional control strategy is developed for the virtual robot, for which inverse kinematics and dynamic equations are formulated and exploited, with appropriate transformations developed for implementation on the continuum robot. Simulations of the virtual robot computed torque control were executed and results indicate that the control method has good trajectory tracking performance. The control algorithm was implemented on a three degree of freedom section of the OctArm continuum manipulator, with decent tracking performance (steady state tracking error of merely 3mm during extension).**

***Index Terms*—Continuum Robot, Control, Kinematics.**

I. Introduction

Continuum or hyper-redundant manipulators belong to a special class of robotic manipulators, which are designed to exhibit behavior similar to biological trunks, tentacles, or snakes [1]. Unlike traditional rigid-link robot manipulators, continuum robot manipulators do not have rigid joints and have many degrees of freedom, and this enables continuum manipulators to have some very useful properties. Continuum manipulators can be compliant, extremely dexterous, flexible, and capable of dynamic adaptive manipulation in highly unstructured environments. These properties of compliant continuum robot manipulators make them uniquely suited for many applications, including search and rescue, underwater operations, and space exploration [2].

Although continuum robots have been prevalent in research for many years [1-3], the development of high-performance control algorithms for these manipulators remains a significant challenge, due to both the complexity and the high degree of uncertainty in their dynamic models. There have been numerous approaches in which researchers have studied various formulations for the control of continuum robot manipulators [4]. Xu et al. [5] developed a computationally efficient torsionally compliant kinematic model of a concentric tube continuum robot. Using this computationally fast technique and deriving the robot's Jacobian, a new position control approach is proposed. Chikhaoui et al. [6] describes theoretical investigations on automation of dual-arm robots constituted of two concentric tube continuum manipulators using motion coordination control. An optimization algorithm is developed to improve triangulation ability of the robot and thus enhance the arms' collaborative operation. Falkenhahn et al. [7] developed a model-based MIMO controller in actuator space, that is based on a spatial dynamic model with one mass point per section. Gravagne et al. [8] discussed the dynamics of a planar continuum backbone section, incorporating a large-deflection dynamic model, formulated a vibration-damping set-point controller, and included experimental results to illustrate the efficacy of the proposed controller. Li et al. [9] developed a model-free method based on an adaptive Kalman filter to accomplish path tracking for a continuum robot using only input pressures and tip position. However, a common element in all these approaches is computational complexity.

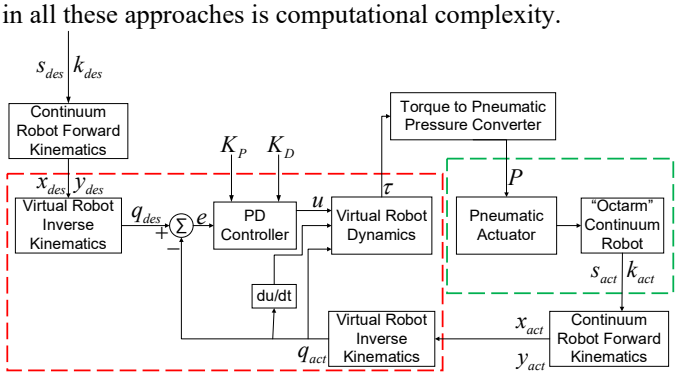

Fig. 1. Block Diagram for Continuum robot control based on virtual robot models

The novel approach to continuum robot control introduced in this paper is motivated by reducing computational complexity. The key innovation is to formulate the overall control strategy 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.

978-1-5090-6684-1/18/$31.00 ©2018 IEEE

Transformations from the desired continuum robot trajectory to the virtual robot, and from the virtual robot control variables to the continuum robot inputs, are developed. This is a completely new approach to the control of continuum robots, to the best of our knowledge. Virtual rigid link robot models have been used to model continuum robot kinematics [10], but this concept has not been extended to controller development previously.

Specifically, in this paper we demonstrate the above approach, from model development to hardware implementation, for control of a single section of a planar continuum robot. The virtual robot used is a serial rigid-link Revolute-Prismatic-Revolute (RPR) joint planar robot with two in-plane rotations and a translation in the same plane. A detailed high-level overview of major system components of this approach is described in Figure 1. First, the desired arc length s and curvature k is fed to continuum robot forward kinematics for desired Euclidean coordinates. Then these desired coordinates are adopted in the virtual RPR robot and controller to acquire the torques and force that brings the continuum robot to a desired coordinate set point. The torques and force are then converted to pneumatic pressure which is directly applied on the continuous backbone of the continuum robot. For modeling, the revolute and prismatic joints are replaced by torsion and extension springs; see Figure 2. The kinematics, dynamics, and controller development established in the following sections are based on this virtual RPR robot and its dynamic behavior.

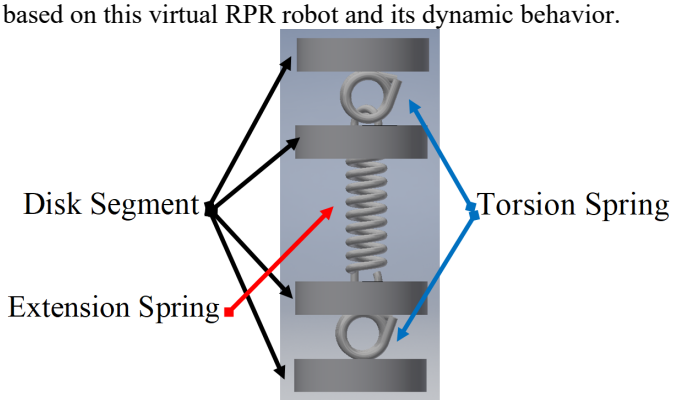

Fig. 2. Continuum robot CAD schematic

In the following sections, first the continuum robot kinematics and necessary transformations, including forward and inverse kinematics, are modeled, referencing the RPR virtual robot. The dynamic model of the virtual robot is established in Section III. Then, based on the dynamic system model, a closed loop computed torque control for the virtual robot is introduced in Section IV. Finally, simulation (Section V) and experimental results (Section VI) are presented, along with related discussion. Conclusions are given in Section VII.

II. Robot Kinematics

Since continuum robots can change their shape at any point along their structure, their models necessarily differ significantly from those of conventional rigid-link robots, where configuration changes can occur only at a finite number of fixed locations along their structure (the joints between the rigid-links). In the following, we review the kinematics of a basic continuum robot element (section) in the plane and relate it to those for the selected virtual robot.

A. Continuum Robot Forward Kinematics

The first (and the most inspired by hardware) approach to continuum robot kinematics strongly exploits the constant curvature sections feature possessed by almost all continuum robots to date [11]. In the plane, a "virtual" three joint rigid-link manipulator, with identical (i.e., coupled) rotations as its first and third joints and a prismatic joint in the middle, can be used to model the kinematic transformation along any constant curvature planar backbone section [10]. Consequently, it is possible to find the corresponding kinematic model, using the conventional Denavit-Hartenberg (D-H) approach [10], for the virtual robot in (1)

$$\left[H_3^0\right]=\begin{bmatrix}\cos(\theta_1+\theta_3) & -\sin(\theta_1+\theta_3) & 0 & -d_2\sin\theta_1\\ \sin(\theta_1+\theta_3) & \cos(\theta_1+\theta_3) & 0 & d_2\cos\theta_1\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \quad (1)$$

The model (1) describes, within the 4 by 4 homogeneous transformation matrix [H], the forward kinematic relationship (3 by 3 orientation, top left of (1), and 3 by 1 translation, top right) between the kinematic variables for the virtual robot (two angles and one length) and task space.

Continuum robot kinematics can now be developed by noting and substituting in the virtual robot kinematics, relationships between the joint variables for the virtual robot and corresponding configuration space variables for the continuous curve. Specifically, ([10], see Figure 3):

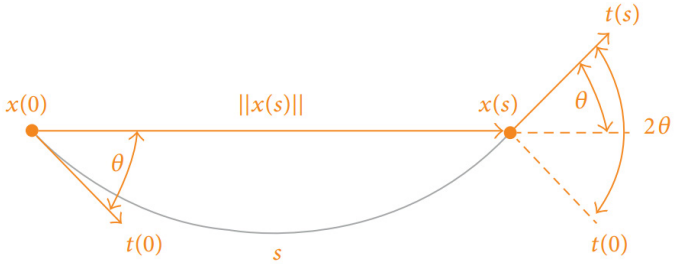

Fig. 3. Geometry of constant curvature section in plane [10]

$$\theta_1=\theta_3=\theta,\quad d_2=\left\|x(s)\right\|,\quad k=\left(\frac{1}{r}\right). \quad (2)$$

We have

$$s=r(2\theta)=\frac{(2\theta)}{k}=\frac{(\theta_1+\theta_3)}{k}\rightarrow(\theta_1+\theta_3)=sk \quad (3)$$

$$\frac{\left\|x(s)\right\|}{2}=\frac{d_2}{2}=r\sin\theta=\frac{\sin\theta}{k}\rightarrow d_2=\frac{2\sin\theta}{k} \quad (4)$$

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

$$\left[H_3^0\right]=\begin{bmatrix} \cos(sk) & -\sin(sk) & 0 & \frac{1}{k}\left(\cos(sk)-1\right) \\ \sin(sk) & \cos(sk) & 0 & \frac{1}{k}\sin(sk) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

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 (arc length and curvature) and task space.

B. Virtual Robot Inverse Kinematics

The inverse kinematics of the continuum robot can be approximated by that of the planar RPR virtual robot. After the continuum robot end-effector's cartesian coordinates are derived from the forward kinematics in (2), the x and y coordinate can then be substituted into the inverse kinematics of the RPR robot to obtain the desired matrix $q_d = [\theta_1 \quad d_2 \quad \theta_3]^T$. The inverse kinematics of the RPR robot can be represented as

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) = \theta_3,\ d_2 = \sqrt{x^2+y^2} \quad (6)$$

III. Virtual Robot Dynamics Model

Consider the virtual Revolute-Prismatic-Revolute (RPR) manipulator shown in Figure 4.

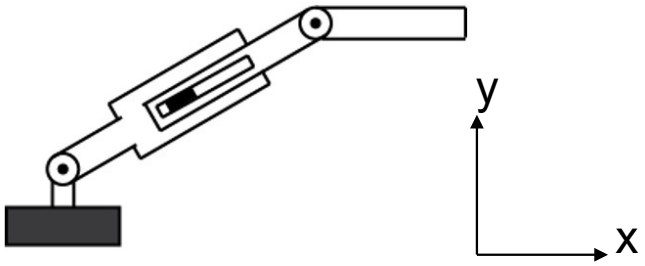

Fig. 4. 3-DoF Revolute-Prismatic-Revolute (RPR) virtual planar robot arm

Let the coordinate system of the base frame (frame 0) be such that z_0 is pointing out of the page and x_0 is pointing to the right. Then, y_0 is pointing towards top in the figure. The joint variables are $q_1 = \theta_1$, $q_2 = d_2$, and $q_3 = \theta_3$. Let the masses of the three links be m_1, m_2, and m_3. Since this is a planar manipulator and rotation is only about the z_0 axis, only the inertia around the vertical axis is relevant; let $I_{1,z}$, $I_{2,z}$, and $I_{3,z}$ denote the moments of inertia of links 1, 2, and 3, respectively, around the axis pointing out of the page (for each link, the moments of inertia are defined relative to a coordinate frame with origin at the center of mass of the link).

If the planar motion of the manipulator is in the horizontal plane, then gravity terms are not relevant. If the planar motion of the manipulator is in the vertical plane, then gravity terms need to be considered.

Let gravity be in the downward direction in the figure (i.e., in the $-y_0$ direction). Let l_{c1} denote the distance from the base (origin of frame 0) to the center of mass of link 1. Let l_1 be the length of link 1. Then, the combined length of links 1 and 2 is $l_1 + q_2$. Also, assume that the linkage between links 1 and 2 is such that when joint 2 actuates, it shifts the center of mass of link 2 by distance q_2. Let the distance from the point where links 2 and 3 meet to the center of mass of link 3 be l_{c3}.

The Euler-Lagrange formulation can be used to find the dynamics of this virtual manipulator. The angular velocity related Jacobian matrices for the three links are:

$$J_{\omega 1} = J_{\omega 2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix},\ J_{\omega 3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad (7)$$

The linear velocity related Jacobian matrices for the three links are:

$$J_{v1} = \begin{bmatrix} -l_{c1}s_1 & 0 & 0 \\ l_{c1}c_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},\ J_{v2} = \begin{bmatrix} -(l_1+q_2)s_1 & c_1 & 0 \\ (l_1+q_2)c_1 & s_1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$
$$J_{v3} = \begin{bmatrix} -(l_1+q_2)s_1 - l_{c3}s_{13} & c_1 & -l_{c3}s_{13} \\ (l_1+q_2)c_2 + l_{c3}c_{13} & s_1 & l_{c3}c_{13} \\ 0 & 0 & 0 \end{bmatrix} \quad (8)$$

where $s_1 = \sin(q_1)$, $c_1 = \cos(q_1)$, $s_{13} = \sin(q_1 + q_3)$, and, $c_{13} = \cos(q_1 + q_3)$. Hence, the matrix $D(q)$ is given by:

$$\begin{aligned} D(q) &= m_1 J_{v1}^T J_{v1} + m_2 J_{v2}^T J_{v2} + m_3 J_{v3}^T J_{v3} \\ &+ J_{\omega 1}^T R_1^0 I_1 \left(R_1^0\right)^T J_{\omega 1} + J_{\omega 2}^T R_2^0 I_2 \left(R_2^0\right)^T J_{\omega 2} \\ &+ J_{\omega 3}^T R_3^0 I_3 \left(R_3^0\right)^T J_{\omega 3} = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \end{aligned} \quad (9)$$

where

$$\begin{aligned} d_{11} &= I_{1,z} + I_{2,z} + I_{3,z} + l_1^2 m_2 + l_1^2 m_3 \\ &+ l_{c1}^2 m_1 + l_{c3}^2 m_3 + m_2 q_2^2 + 2 l_1 m_2 q_2 + 2 l_1 m_3 q_2 \\ &+ 2 l_1 l_{c3} m_3 \cos(q_3) + 2 l_{c3} m_3 q_2 \cos(q_3) \end{aligned} \quad (10)$$

$$d_{12} = d_{21} = -l_{c3} m_3 \sin(q_3) \quad (11)$$

$$\begin{aligned} d_{13} = d_{31} &= I_{3,z} + l_{c3}^2 m_3 + l_1 l_{c3} m_3 \cos(q_3) \\ &+ l_{c3} m_3 q_2 \cos(q_3) \end{aligned} \quad (12)$$

$$d_{22} = m_2 + m_3 \quad (13)$$

$$d_{23} = d_{32} = -l_{c3} m_3 \sin(q_3) \quad (14)$$

$$d_{33} = I_{3,z} + l_{c3}^2 m_3 \quad (15)$$

As described above, since the rotation of all the links is only about the z_0 axis, only the moments of inertia about the axis pointing out of the page are relevant (i.e., $I_{1,z}$, $I_{2,z}$, $I_{3,z}$).

The Christoffel symbols c_{ijk} become

$$c_{ijk} = \frac{1}{2}\left\{\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right\} \quad (16)$$

For $i = 1,2,3; j = 1,2,3; k = 1,2,3$, and writing the matrix $C(q,\dot{q})$ with its $(k,j)^{th}$ element being

$$c_{kj}\sum_{i=1}^{n}c_{ijk}(q)\dot{q}_i \tag{17}$$

we obtain

$$C(q,\dot{q})=\begin{bmatrix} C_{11}(q,\dot{q}) & C_{12}(q,\dot{q}) & C_{13}(q,\dot{q}) \\ C_{21}(q,\dot{q}) & C_{22}(q,\dot{q}) & C_{23}(q,\dot{q}) \\ C_{31}(q,\dot{q}) & C_{32}(q,\dot{q}) & C_{33}(q,\dot{q}) \end{bmatrix} \tag{18}$$

where

$$C_{11}(q,\dot{q})=\dot{q}_2\begin{pmatrix} l_1m_2+l_1m_3+m_2q_2+m_3q_2 \\ +l_{c3}m_3\cos(q_3) \end{pmatrix} -l_{c3}m_3\dot{q}_3\sin(q_3)(l_1+q_2) \tag{19}$$

$$C_{12}(q,\dot{q})=\dot{q}_1\begin{pmatrix} l_1m_2+l_1m_3+m_2q_2 \\ +m_3q_2+l_{c3}m_3\cos(q_3) \end{pmatrix} \tag{20}$$

$$C_{13}(q,\dot{q})=-l_{c3}m_3\sin(q_3)(l_1+q_2)(\dot{q}_1+\dot{q}_3) \tag{21}$$

$$C_{21}(q,\dot{q})=-\dot{q}_1\begin{pmatrix} l_1m_2+l_1m_3+m_2q_2+m_3q_2 \\ +l_{c3}m_3\cos(q_3) \end{pmatrix}-l_{c3}m_3\dot{q}_3\cos(q_3) \tag{22}$$

$$C_{22}(q,\dot{q})=0 \tag{23}$$

$$C_{23}(q,\dot{q})=-l_{c3}m_3\cos(q_3)(\dot{q}_1+\dot{q}_3) \tag{24}$$

$$C_{31}(q,\dot{q})=l_{c3}m_3\dot{q}_2\cos(q_3)+l_{c3}m_3\dot{q}_1\sin(q_3)(l_1+q_2) \tag{25}$$

$$C_{32}(q,\dot{q})=l_{c3}m_3\dot{q}_1\cos(q_3) \tag{26}$$

$$C_{33}(q,\dot{q})=0 \tag{27}$$

The potential energy of the manipulator is given by:

$$P=m_1gl_{c1}\sin(q_1)+m_2g(l_1+q_2)\sin(q_1) +m_3g\left(\sin(q_1)(l_1+q_2)+l_{c3}\sin(q_1+q_3)\right) \tag{28}$$

Hence,

$$g(q)=\begin{bmatrix}\frac{\partial P}{\partial q_1} & \frac{\partial P}{\partial q_2} & \frac{\partial P}{\partial q_3}\end{bmatrix}^T=\begin{bmatrix}g_{11} & g_{12} & g_{13}\end{bmatrix}^T \tag{29}$$

where

$$g_{11}=m_1gl_{c1}\cos(q_1)+m_2g(l_1+q_2)\cos(q_1) +m_3g\left(\cos(q_1)(l_1+q_2)+l_{c3}\cos(q_1+q_3)\right) \tag{30}$$

$$g_{12}=m_2g\sin(q_1)+m_3g\sin(q_1) \tag{31}$$

$$g_{13}=m_3gl_{c3}\cos(q_1+q_3) \tag{32}$$

The dynamic equations of the manipulator are given by:

$$D(q)\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\\ \ddot{q}_3\end{bmatrix}+C(q,\dot{q})\dot{q}+g(q)=\begin{bmatrix}\tau_1\\ f_2\\ \tau_3\end{bmatrix} \tag{33}$$

where τ_1 is the applied torque at the first joint (revolute), f_2 is the applied force at the second joint (prismatic), and τ_3 is the applied torque at the third joint (revolute).

IV. Overall Control Method

We seek and exploit simple, relatively computationally inexpensive control methods used in (rigid-link) robot control systems [12] to design the controller in the virtual robot coordinates.. Classical control and intelligent control methods are widely used in the robot industry. Each control method has advantages and disadvantages. However, the main aim for the system is to provide robustness, stability and high frequency updates. In this work, we adopt the computed torque (feedback linearization plus PD control, see Figure 5) approach for the virtual robot, with the sensing and actuation transformed from and to the continuum robot, respectively.

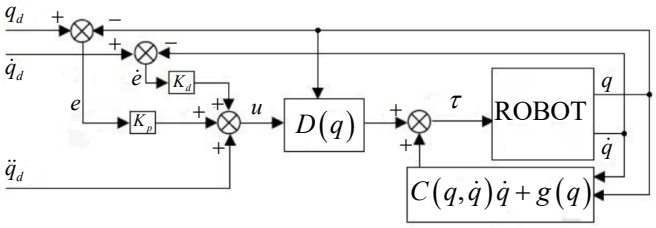

Fig. 5. Classical control block diagram of the robot arm

Conventional PD control [13] is the most popular core control method in many robot implementations because of its steady state and transient response performance in time-invariant systems. In classical pure PD control, the chosen parameters, K_p and K_d remain constant during the process. Therefore, such a controller is inefficient because the controller contains ambiguity when environmental conditions or dynamics change. In addition, it is inefficient because of time delays and nonlinearity conditions. Hence, we include the dynamics to linearize prior to the PD control.

The dynamic model of the virtual robot arm is given in (33), In this equation, τ, $D(q)$, $C(q,\dot{q})$, $G(q)$, and n are expressions for the $n\times 1$ dimensional joint torque, $n\times n$ dimensional inertia matrix, $n\times 1$ dimensional Coriolis and centrifugal vector, $n\times 1$ dimensional gravity vector, and the degrees of freedom of the robot, respectively. The errors of the robot link variables are

$$e=q_d-q,\ \dot{e}=\dot{q}_d-\dot{q},\ \ddot{e}=\ddot{q}_d-\ddot{q} \tag{34}$$

where $e,\dot{e},\ddot{e}$ expresses the position, velocity and acceleration error vectors and $q_d,\dot{q}_d,\ddot{q}_d$ expresses the desired position, velocity and acceleration of the link variables. The torques required for each joint of the virtual robot arm are calculated from (33) and the errors from (34). The linearization is achieved as follows

$$\tau=D(q)(\ddot{q}_d-u)+C(q,\dot{q})\dot{q}+g(q) \tag{35}$$

The control signal that is obtained from (35) is expressed as follows

$$u=\ddot{q}_d+D^{-1}(q)\left[C(q,\dot{q})\dot{q}+g(q)-\tau\right] \tag{36}$$

If the signal u is selected as the PD feedback controller, the torque value of each joint will be obtained from (37) and (38).

$$u=-K_d\dot{e}-K_pe \tag{37}$$

$$\tau=D(q)\left(\ddot{q}_d+K_d\dot{e}+K_pe\right)+C(q,\dot{q})\dot{q}+g(q) \tag{38}$$

where K_d is the derivative gain and K_p is the proportional gain.

The overall controller of the virtual robot is shown in Figure 5. The PD coefficients of the system were tuned experimentally, and the ideal gain values were used. The input desired trajectory was represented in terms of Cartesian coordinates x and y, and was calculated from the continuum robot arc length s and

curvature k using the forward kinematics discussed in section II. Subsequently the virtual robot variables: rotation θ_1, θ_3 and translation d_2 were derived from the inverse kinematics in section II and fed into the control system as a desired reference input signal. Their derivatives and double derivatives were calculated and input to the controller. The output of the controller, u, is then used to establish the torque signal along with systems $D(q)$, $C(q,\dot{q})$, and $g(q)$ matrix. The torque was then converted and applied to the physical robot system which feeds back the current continuum robot shape, subsequently converted to virtual robot rotation and translation signals input to the PD controller to form the error and drive the control action.

V. Simulation Results

Simulations of the virtual robot computed torque control were executed in the Simulink environment given the input of the system is a reference signal of the arc length s and curvature k. Feeding into the forward kinematics to form Cartesian coordinates x and y, the reference signal illustrated here is a chirp signal in which the frequency increases with time shown in Figure. 6.

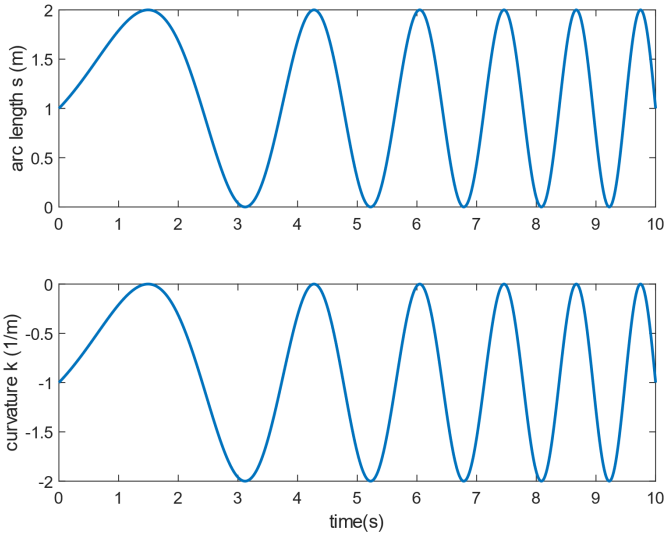

Fig. 6. Desired continuum robot arc length s and curvature k input to the simulation of the virtual robot computed torque control

The resulting output torque for the virtual robot is illustrated in Figure. 7. The first and third subplots are the rotational torque of the first and last virtual torsion springs τ_1 and τ_3 while the second plot is for the translation force f_2 of the virtual extension spring located in between the two virtual torsion springs. The torsion coefficient can be calculated by selecting a torque of a given time and identifying the twist angle

$$k_t = -\frac{\tau}{\theta} \tag{39}$$

The spring constant of the extension spring can be calculated using the same method:

$$k_e = -\frac{f_2}{d_2} \tag{40}$$

where f_2 is the force exerted by extension spring illustrated in Figure 7 and d_2 is the elongation of the virtual spring.

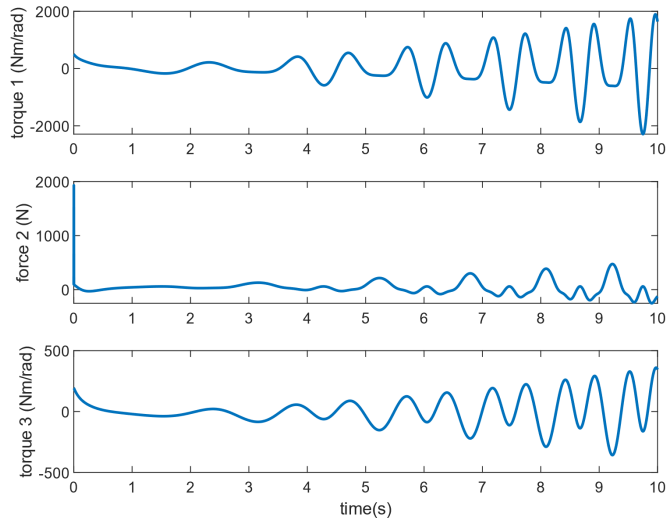

Fig. 7. Torques and force applied to the springs: (a) torque applied to torsion spring τ_1 , (b) force applied to extension spring f_2, (c) torque applied to torsion spring τ_3

The comparison between desired and actual x and y coordinates of the continuum robot end effector is depicted in Figure 8. There is a large overshoot observed when the end effector is attempting to reach the first desired x location, but then the system becomes stabilized. The overshoot of the y coordinate is zero and it also become stable after the first chirp signal peak.

The desired and actual $\theta_1, \dot{\theta}_1, d_2, \dot{d}_2, \theta_3, \dot{\theta}_3$ of the virtual robot can be observed in Figure. 9. All signals eventually reach a high tracking precision, indicating that the controller accomplished its task of bringing the extension and torsion springs to the desired position and velocity. The derivative terms all have overshoot issues that can be neglected in the practical implementation (see next section).

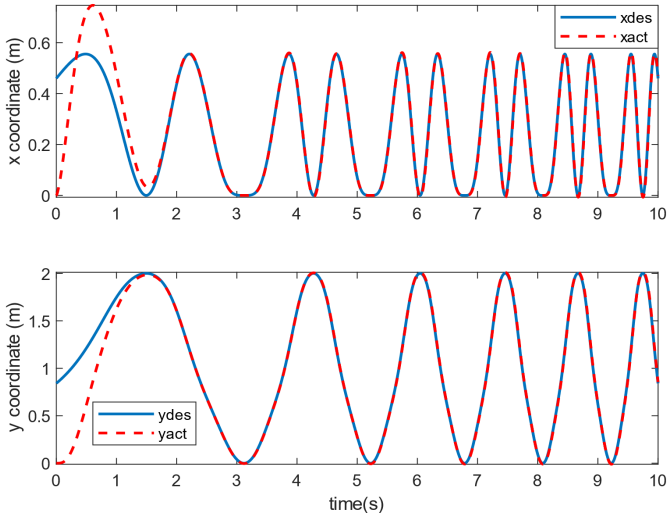

Fig. 8. Desired and actual continuum robot end-effector X and Y coordinate to the simulation of the virtual robot computed torque control

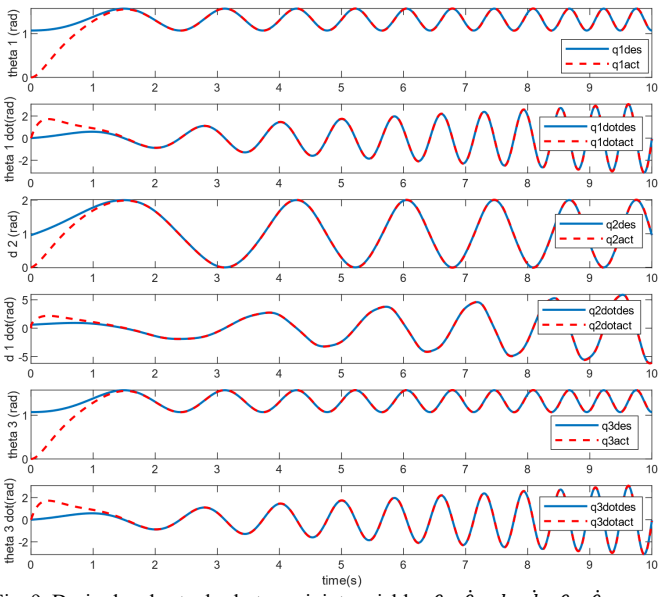

Fig. 9. Desired and actual robot arm joint variables θ_1, $\dot{\theta}_1$, d_2, $\dot{d}_2$, θ_3, $\dot{\theta}_3$

VI. Experimental Results

1The controller was implemented on the tip section of the OctArm continuum manipulator [14]. The OctArm, pictured in Figure 10, is a pneumatically actuated, three section, nine degree of freedom (DoF) continuum manipulator. Each section is capable of bending in any direction (curvature k and direction φ) and extending (arc length s), providing three DoF for each section. The tip section of the device (the right-most section in Figure 10) is comprised of three McKibben extension muscles [15] arranged radially at 120° intervals.

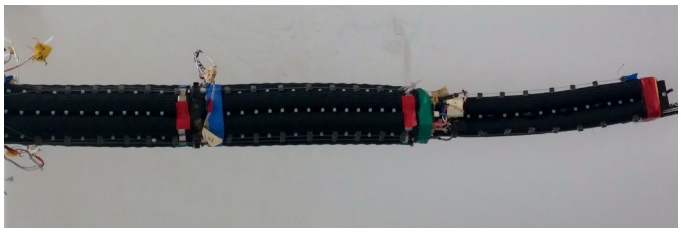

Fig. 10. The OctArm Manipulator

A set of experiments utilizing the OctArm and the described model were implemented. The model and controller were implemented in MATLAB/Simulink [16]. Interfacing with the OctArm was accomplished using two Quanser Q8-usb data acquisition boards [17]. 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 controller output torques and force are calculated, they are converted to pneumatic pressure in voltage form which then can be applied onto the three McKibben extension muscles at the tip section.

A. Extension

The first experiment is pure extension of the OctArm continuum manipulator. In this experiment, the system is fed with arc length s being a sinusoid with an amplitude 0.03m of and a frequency of 0.08Hz and curvature k being 0m^{-1}. For the extension experiment, the calculated extension force f_2 that results from the model is equally applied to the three muscles to achieve balanced pure extending movement. The section desired and actual arc length are presented in Figure 11. During the experiment, the OctArm initiated from its natural unpressurized length of 0.34m and immediately converged to the desired arc length with minor error in the crest of the sine wave. The actual arc length settles relatively fast and no obvious overshoot or oscillations are detected. The arc length error plot illustrated in Figure 12 shows that the control algorithm implemented on the extension of OctArm only outputs an error of ± 3 mm which is considered within a reasonable range for this robot. The extension force applied to the OctArm through the pneumatic actuators can also be observed in Figure 12. If the OctArm is considered as an extension spring during this experiment, then the spring constant for the spring can be calculated from (40) after knowing the extension force amplitude of 28.71N from Figure 12 and arc length amplitude of 0.03m. The spring constant k_e is approximately equal to 957N/m.

B. Bending

In the second experiment, a bending test on the OctArm is carried out. The objective of the experiment is to have the OctArm maintain a constant arc length s at 0.395m while bending the continuum robot section to track a sinusoid curvature k with an amplitude of 0.2m^{-1} and frequency of 0.08Hz. Figure 13 depicts the bending experiment, showing the tip section of the OctArm and the travel between the maximum and minimum curvature values.

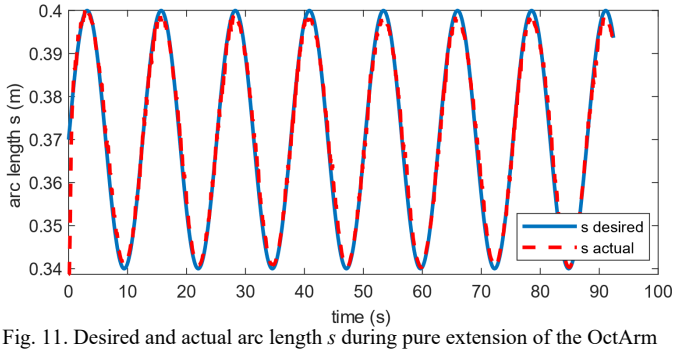

Fig. 11. Desired and actual arc length s during pure extension of the OctArm continuum manipulator

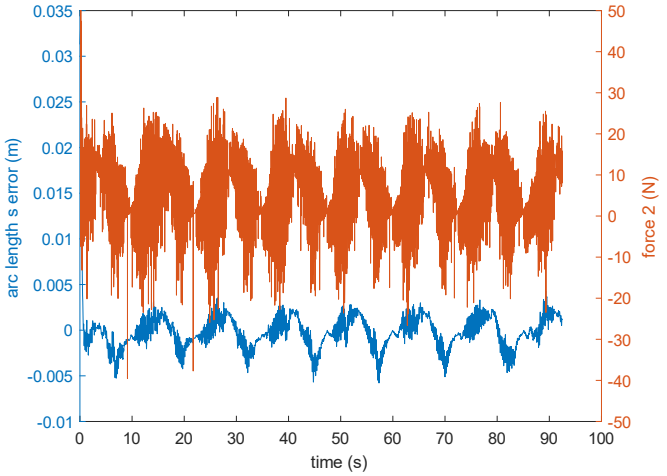

Fig. 12. Arc length s error and PD controller extension force during pure extension of the OctArm continuum manipulator

Fig. 13. Oscillating curvature experiment

To achieve bending, the average of τ_1 and τ_3 is calculated first as $\Delta\tau$. Then the applied pneumatic pressure onto the two McKibben extension muscles at the back of the tip section (pictured on the right hand side of the OctArm in Figure 13) is calculated as $P_1 = k_p(f_2 + \Delta\tau)$ and the applied pneumatic pressure onto the one extension muscle at the front of the tip section (pictured on the left hand side in Figure 13) is calculated as $P_2 = k_p(f_2 - \Delta\tau)$ where k_p is the conversion gain from torque to pressure. The difference of pressure given to two distinct sets of extension muscles at front and back will generate a bending effect of constant curvature that matches the continuum robot kinematics model.

The section arc length and curvature are both depicted in Figure 14 with the desired and actual values compared. The rise time and settling time for the arc length is 64s and 88s respectively and no overshoot is observed. The curvature, on the other hand, displays a relatively fast response compared to arc length, with an overshoot of 8%. The reason for a slower response from the arc length is posited to be the following: during bending, the continuum robot is attempting to achieve the desired sinusoid curvature that could lead to a shrinkage of the robot that contradicts and neutralize the extension force. The error between desired set-point of arc length and curvature and actual values can be observed in Figure 15.

The comparison between desired and actual x and y coordinates of the continuum robot end effector is depicted in Figure 16. There is a large undershoot observed when the end effector is attempting to reach the desired x location, but the system ultimately becomes stabilized. The overshoot of the y coordinate is relatively small, and it also becomes stabilized after three cycles of the sine wave.

The desired and actual θ_1, d_2, θ_3 of the virtual robot model can be observed in Figure 17. The values θ_1 and θ_3 converge to the desired set-point successfully thanks to the overwhelming tracking performance of the curvature whereas d_2 has inferior performance due to the long rise time and settling time of the arc length control. The resulting output torques for the virtual robot during bending are illustrated in Figure 18.

After successfully employing the control method of the virtual discrete-jointed robot model onto the OctArm, both

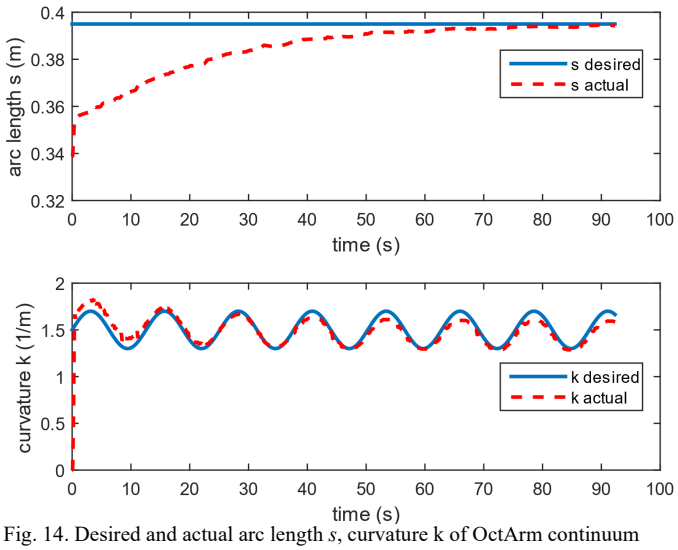

Fig. 14. Desired and actual arc length *s*, curvature k of OctArm continuum manipulator during oscillating curvature experiment

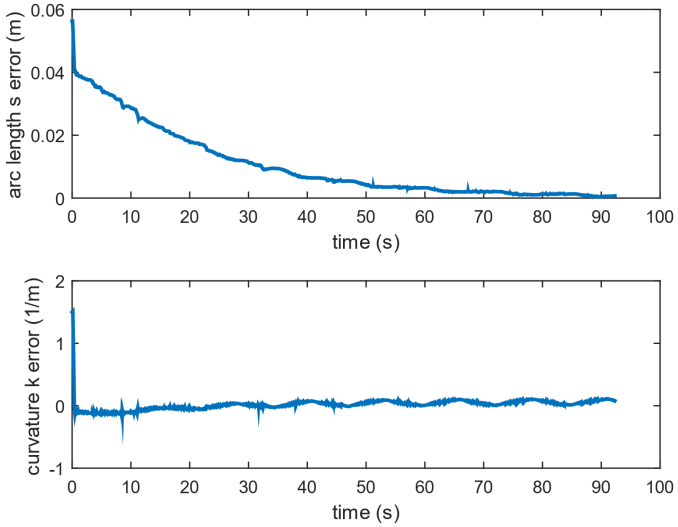

Fig. 15. Arc length *s* and curvature k error of OctArm continuum manipulator during oscillating curvature experiment

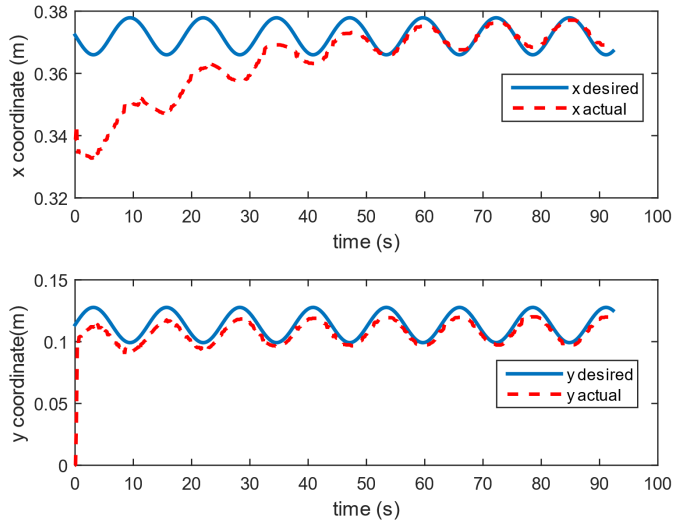

Fig. 16. Desired and actual X, Y coordinate of OctArm continuum manipulator end-effector during oscillating curvature experiment

extension and bending tests deliver relatively ideal tracking performance. The results from both test procedures show decent consistency with the simulation results of computed torque

approach for the virtual robot, and good potential for use in continuum controller implementation.

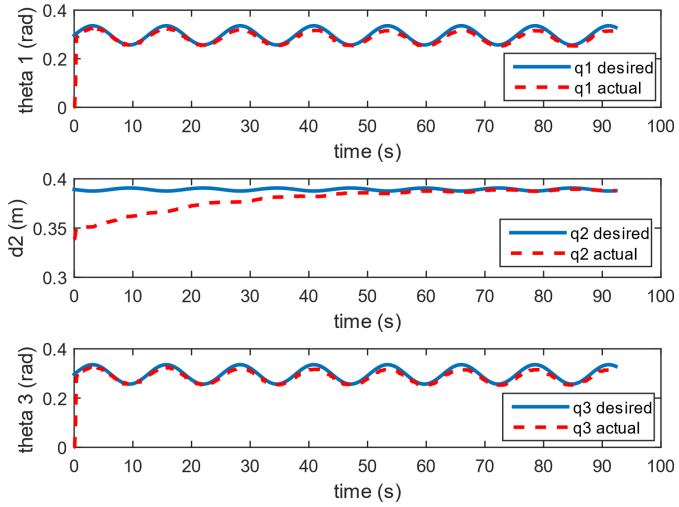

Fig. 17. Desired and actual θ_1, d_2, θ_3 of OctArm continuum manipulator during oscillating curvature experiment

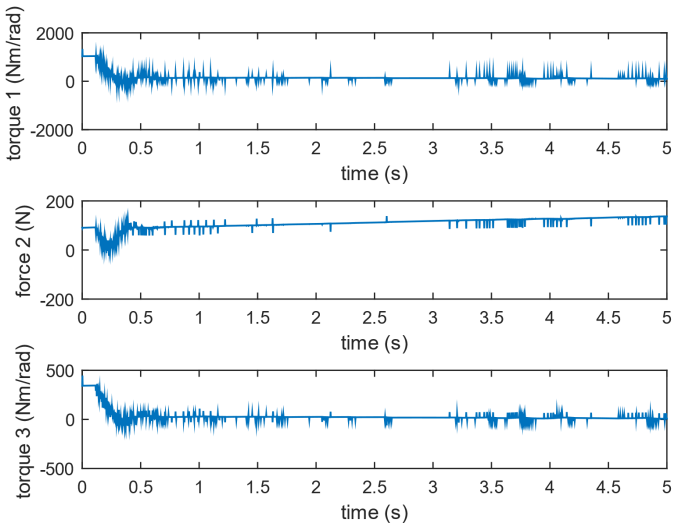

Fig. 18. Torques and force applied to the OctArm continuum manipulator during during oscillating curvature experiment: (a) torque applied to torsion spring τ_1 , (b) force applied to extension spring f_2, (c) torque applied to torsion spring τ_3

VII. Conclusion

This paper introduces a new and novel approach to the control of continuum robots. The main innovation is the use of a virtual, conventional rigid-link robot model, in whose coordinates the controller is developed, to generate the real-time control inputs for the continuum robot. The key advantage of the control approach presented in this paper is that it can be implemented very efficiently. The proposed controller was shown to provide reasonable performance in both simulations and experiments, without the need for excessive online or pre-computation.

Acknowledgments

This work is supported in part by the U.S. National Science Foundation under grants IIS-1527165 and IIS-1718075, in part by NASA under contract NNX12AM01G, and in part by a NASA Space Technology Research Fellowship, contract 80NSSC17K0173.

References

[1] G. Robinson and J. B. C. Davies, “Continuum Robots - A State of the Art,” *Proceedings IEEE International Conference on Robotics and Automation*, Detroit, Michigan, pp. 2849-2854, 1999.

[2] D. Trivedi, C.D. Rahn, W.M. Kier, and I.D. Walker, “Soft Robotics: Biological Inspiration, State of the Art, and Future Research”, *Applied Bionics and Biomechanics*, vol.5, no.2, pp. 99-117, 2008.

[3] R.J. Webster III and B.A. Jones, “Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review”, *International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661-1683, November 2010.

[4] T.G. Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control Strategies for Soft Robotic Manipulators: A Survey", *Soft Robotics*, Vol. 5, No. 2, pp. 149-163, 2018.

[5] R. Xu, A. Asadian, A. S. Naidu, and R. V. Patel, "Position Control of Concentric-Tube Continuum Robots using a Modified Jacobian-Based Approach," *IEEE International Conference on Robotics and Automation,* Karlsruhe, Germany, May 2013.

[6] M. T. Chikhaoui, J. Granna, J. Starke, and J. Burgner-Kahrs, "Toward Motion Coordination Control and Design Optimization for Dual-Arm Concentric Tube Continuum Robots," *IEEE Robotics and Automation Letters,* vol. 3, no. 3, pp. 1793-1800, 2018.

[7] V. Falkenhahn, A. Hildebrandt, R. Neumann, and O. Sawodny, "Model-based Feedforward Position Control of Constant Curvature Continuum Robots Using Feedback Linearization," 2015 IEEE Conference on Robotics and Automation, pp. 762-767, May 2015, Seatle, WA.

[8] I. A. Gravagne, C. D. Rahn, and I. D. Walker, "Large Deflection Dynamics and Control for Planar Continuum Robots," *IEEE/ASME Transactions on Mechatronics,* vol. 8, no. 2, pp. 299-307, 2003.

[9] M. Li, R. Kang, D. T. Branson, and J. S. Dai, "Model-Free Control for Continuum Robots Based on an Adaptive Kalman Filter," *IEEE/ASME Transactions on Mechatronics,* vol. 23, no. 1, pp. 286-297, 2018.

[10] M. W. Hannan and I. D. Walker, “Kinematics and the Implementation of an Elephant's Trunk Manipulator and Other Continuum Style Robots,” *Journal of Robotic Systems*, vol. 20, no. 2, pp. 45-63, Feb. 2003.

[11]I. D. Walker, "Continuous Backbone “Continuum” Robot Manipulators," *ISRN Robotics,* vol. 2013, pp. 1-19, 2013.

[12]J. Shah, S. S. Rattan, and B. C. Nakra, "Dynamic Analysis of Two Link Robot Manipulator for Control Design Using Computed Torque Control," *International Journal of Research in Computer Applications and Robotics,* vol. 3, no. 1, pp. 52-59, January 2015.

[13]Z. Ortatepe and O. Parlaktuna, "Two Dof Robot Control with Fuzzy Based Neural Networks," *Aanadolu University Journal of Science and Techology A - Applied Sciences and Engineering,* vol. 18, no. 4, pp. 819-830, 2017.

[14] W. McMahan, M. Pritts, V. Chitrakaran, D. Dienno, M. Grissom, B. Jones, M. Csencsits, C.D. Rahn, D. Dawson, and I.D. Walker, “Field Trials and Testing of “OCTARM” Continuum Robots”, *Proceedings IEEE International Conference on Robotics and Automation*, pp. 2336-2341, 2006, Orlando, FL.

[15] I. Walker, D. Dawson, T. Flash, F. Grasso, R. Hanlon, B. Hochner, W. Kier, C. Pagano, C. Rahn, and Q. Zhang, “Continuum robot arms inspired by cephalopods,” in Proc. SPIE Conf. Unmanned Ground Veh. Tech., Orlando, FL, 2005, pp. 303–314.

[16] MathWorks, 2018. Simulation and model based design, April. http://www.mathworks.com/products/simulink.

[17] Quanser, 2018. Q8-usb data acquisition board, April. http://www.quanser.com/products/q8-usb.