A Consensus Strategy for Decentralized Kinematic Control of Multi-Segment Soft Continuum Robots

Amir Salimi Lafmejani, Hamed Farivarnejad, Azadeh Doroudchi, and Spring Berman

***Abstract*— This paper proposes a novel decentralized approach to kinematic control of soft segmented continuum robots based on a consensus strategy. The robots under consideration deform in a plane according to a multi-segment Piecewise Constant Curvature (PCC) kinematic model in which each segment is represented as an equivalent rigid-link Revolute-Prismatic-Revolute (RPR) mechanism. In our approach, we assume that each segment of the robot is equipped with sensors to measure joint variables in its local coordinate frame and can communicate with its two adjacent segments. Our consensus-based decentralized control strategy provides an alternative to conventional control methods, which solve the inverse kinematic problem by using computationally intensive numerical methods to calculate the robot's Jacobian matrix at each time instant. We investigate the stability and convergence properties of proposed controllers for position regulation and trajectory tracking tasks and provide theoretical guarantees on the controllers' performance. We evaluate the controllers in simulation for scenarios in which the robot's tip must reach a certain position or follow a specified trajectory. We compare the performance of the position regulator for different controller gains, and we find that a simulated 15-link robot can track a complex reference trajectory with an average root-mean-square error of only 0.16% of the robot's initial length.**

I. Introduction

A. Motivation

Decentralized control approaches can be used to overcome limitations of centralized control approaches when applied to large-scale systems, such as high computational complexity, delays, uncertainties, and a lack of robustness [1]. Recently, decentralized approaches have attracted the attention of robotics researchers as reliable methods for kinematic and dynamic control of robotic systems. These approaches have been extensively employed in the control of distributed robotic systems such as groups of ground robots [2] and Unmanned Aerial Vehicles (UAVs) [3], as well as hyper-redundant soft [4] and continuum robots [5], [6].

Continuum robots in particular have been developed for a variety of uses, including medical procedures (e.g., steerable needles), remote inspection, search-and-rescue, and space applications [8]. The hyper-redundancy of continuum robots creates many challenges for conventional centralized control strategies, e.g., the large number of solutions for the Inverse Kinematic (IK) problem, which motivates the use of alternative control approaches. Existing alternative methods that use centralized control have employed numerical methods [9] and the pseudo-inverse Jacobian method [10], [11]. However, these approaches are limited by their high computational complexity, which make them inefficient for real-time applications. Moreover, centralized control of a continuum robot requires the measurement of the robot's joint variables represented in the global coordinate frame, which makes them unsuitable for autonomous control applications. The computational effort of implementing such controllers is likely to rise as the number of joints in the robot increases.

This work was supported by Office of Naval Research Award N00014-17-1-2117 and by the Arizona State University Global Security Initiative.

A. Salimi Lafmejani and A. Doroudchi are with the School of Electrical, Computer and Energy Engineering, Arizona State University (ASU), Tempe, AZ, 85287 {asalimil, adoroudc}@asu.edu. H. Farivarnejad and S. Berman are with the School for Engineering of Matter, Transport and Energy, ASU, Tempe, AZ 85287 {hamed.farivarnejad, spring.berman}@asu.edu.

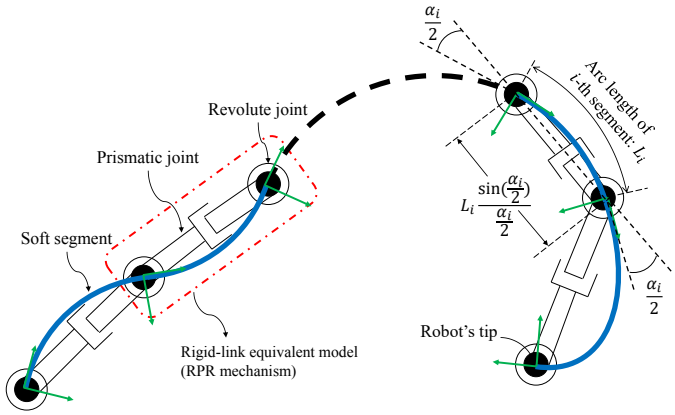

Fig. 1. Schematic of a continuum robot composed of soft segments, along with its equivalent model as a serial configuration of multiple rigid-link RPR mechanisms. Local coordinate frames (Frenet-Serret frames [7]) are defined at the two ends of each soft segment.

B. Related Work

The discrete method [12] is widely employed for kinematic modeling of soft continuum robots. In this method, the robot is divided into multiple segments that each deform in the shape of a circular arc, which is referred to as the Constant Curvature (CC) assumption. Although soft continuum robots generally can deform into shapes with variable curvature, many existing soft robots are designed to conform to the CC assumption [13], [14], which enables the derivation of closed-form kinematics and Jacobian formulations [7]. In the discrete method, each soft curved segment is modeled as an equivalent rigid-link RPR mechanism, as shown in Fig. 1.

Several methods have been proposed for kinematic control of segmented continuum robots. The fitting algorithm was investigated in [15]. In [16], a modular control scheme was proposed to control the configuration of a segmented contin-

uum robot, aiming at reducing the computational load of the fitting algorithm by dividing the robot into multiple modules. However, this work assumed that the desired backbone curve configuration is given, which addresses the kinematic control of the robot in a centralized fashion. Another widely-used method for kinematic control of segmented continuum robots is the pseudo-inverse Jacobian method, which utilizes the Moore-Penrose pseudo-inverse of the robot's Jacobian matrix to generate minimum-magnitude joint velocities for control [10]. The main drawback of this centralized method is that the hyper-redundancy of the robot results in a large rectangular Jacobian matrix whose pseudo-inverse is time-consuming to compute, preventing the method from being efficient in practice.

Decentralized control methods can be used to avoid the aforementioned limitations of centralized control approaches for kinematic control [17], cooperative control [18], and fault-tolerant control [19] of multi-segment continuum robots. Moreover, decentralized control approaches can significantly reduce the computational complexity of the control strategy. Recently, decentralized approaches have been employed for the control of multi-segment continuum robots with soft segments. In [20], a decentralized control mechanism with local feedback was developed for a multi-segment millipede-like robot. In [21], a soft multi-segment shape-changing robot was developed with integrated sensing, actuation, and process modules, and a distributed controller was designed for shape control of the robot. A reinforcement learning-based approach was proposed in [22] to control serpenoid locomotion of a snake-like continuum robot. Furthermore, a decentralized method for shape control of a rigid-link continuum manipulator was proposed in [23] and verified in simulations and experiments. In [17], the authors were inspired by the analogy between the heat and wave equations to propose a decentralized control approach for hyper-redundant robots. The work [24] proposed a modular decentralized control approach for a general N-segment single-DoF continuum robot that exploited its stable configurations. In [25], a decentralized control approach was presented for a 1D soft robot arm composed of segments with local sensing, actuation , and control, and was validated in simulation. Novel morphological observation and decentralized control approaches were presented in [26] for passive shape adaptation, geometrical disturbance rejection, and task space anisotropic stiffness regulation of a 3D-printable thermoactive helical interface on a continuum manipulator.

C. Contributions

Most consensus protocols in robotics are designed for multiple mobile robots, such as in [27], [28]. In these consensus strategies, the robots reach an agreement on the value of an information state through computations on their own state and those communicated by other robots [29]. In this paper, we define a consensus protocol for a multi-segment continuum robot by considering the segments' local measurements of their own configurations as the shared information states. The segments communicate these measurements to adjacent segments, as illustrated in Fig. 2. Similar inter-segment communication is present in the decentralized nervous system in octopus arms; this nervous system is organized as segments along the length of each arm, and sensorimotor information is propagated between neighboring segments [30]. Thus, our decentralized, consensus-based strategy for kinematic control can be viewed as a bio-inspired control strategy for soft multi-segment continuum robots.

In this paper, we present a multi-segment kinematic model for a soft continuum robot in which each segment is modeled as a rigid RPR mechanism, described in Section II. As depicted in Fig. 2, we assume that each segment can measure and communicate to adjacent segments the position of its equivalent rigid-link prismatic joint and the relative orientation between this link's two revolute joints in its local coordinate frame. In Section III, we propose decentralized controllers that utilize a consensus protocol on these measurements for the objectives of driving the robot's tip to reach a target position and to track a reference trajectory. Using this novel controller design, it is not necessary to calculate the Jacobian matrix of the multi-segment continuum robot, which in turn reduces the complexity of the controller implementation and improves its efficiency. We analyze the stability and convergence properties of the controllers in Section IV for the general case of a robot with N segments. In Section V, we validate our position regulation and trajectory tracking controllers in simulation on 5-segment and 15-segment robots and illustrate the effect of the controller gain.

II. Rigid-Link Model of Soft Continuum Robot

The kinematics of the soft continuum robot are discussed in this section. The robot is composed of a set of segments connected to each other in a series configuration. The soft segments of the robot are replaced by equivalent rigid-link RPR mechanisms [31].

Figure 1 depicts a soft continuum robot with soft segments, which is equivalently modeled with multiple rigid-link RPR mechanisms connected in a series. To clarify the details of the model, Figs. 2 and 3 show the soft segmented robot and its equivalent rigid-link robot, respectively. Figure 2 illustrates a planar segmented soft continuum robot with N bending segments, each conforming to the constant-curvature (CC) assumption. The kinematic model of the continuum robot is defined by the kinematic equations of the multi-segment N-RPR rigid-link robot in Fig. 3. Since we assume that each segment of the robot is equipped with local sensors and actuators, it is able to measure the position of its prismatic joint and relative rotations of its revolute joints in its local coordinate frame. Furthermore, the i-th segment can communicate these local measurements to the adjacent $(i-1)$-th and $(i+1)$-th segments, as shown in Fig. 2.

As shown in Fig. 1, the angular difference between the tangential local coordinate frames attached to the base and end-effector of the i-th segment and the orientation of the equivalent RPR rigid-link mechanism is defined as $\frac{\alpha_i}{2}$. Furthermore, we denote the arc length of the i-th soft segment

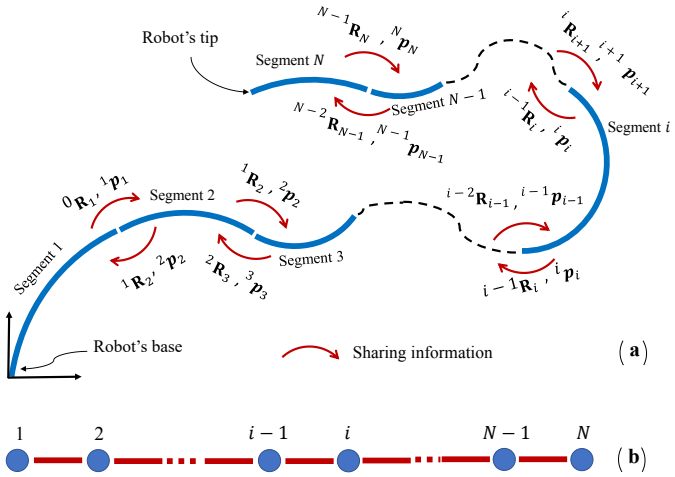

Fig. 2. Schematic of information propagation between segments of the continuum robot. (a) Each segment $i \in \{1, ..., N\}$ communicates with its adjacent segment(s) in order to share its measurements of the position vector ${}^{i}\boldsymbol{p}_i$ and rotation matrix ${}^{i-1}\boldsymbol{R}_i$, which are represented in its local coordinate frame. (b) Graph of the robot's communication network, where each blue node represents a segment and each red edge represents a bidirectional communication channel.

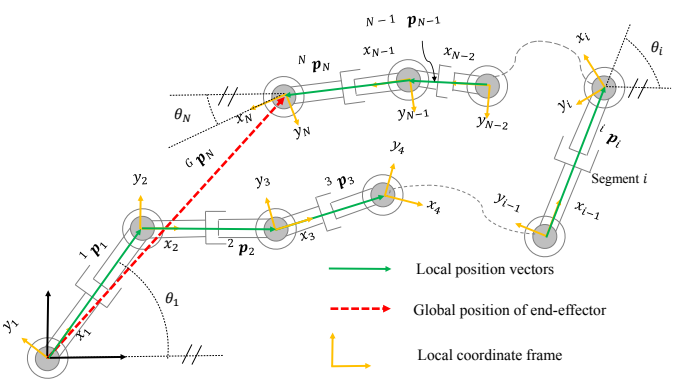

Fig. 3. Illustration of an equivalent N-RPR rigid-link model for the soft continuum robot in Fig. 2. The x-axis of each local coordinate frame is aligned with the corresponding prismatic joint.

by L_i. Accordingly, as in [31], the position vector that is aligned with the prismatic joint of the i-th segment can be represented in the segment's local coordinate frame as the following vector ${}^{i}\boldsymbol{p}_i$:

$$ {}^{i}\boldsymbol{p}_i = \begin{bmatrix} L_i \dfrac{\sin(\alpha_i)}{\alpha_i} & L_i \dfrac{1-\cos(\alpha_i)}{\alpha_i} & 0 \end{bmatrix}^{\mathrm{T}}. \tag{1} $$

Thus, given the arc parameters L_i and α_i of each soft segment, the position vector ${}^{i}\boldsymbol{p}_i$ of the equivalent RPR mechanism is readily obtained.

III. Controller Design

In this section, we propose a novel decentralized method for kinematic control of soft segmented continuum robots. This control method utilizes a consensus strategy among the segments of the robot, which can communicate measurements to each other according to a chain network topology.

A. Position Regulation Controller

Here, we define a control law that drives the tip of the robot to a target 3D position in the global frame G, denoted by ${}^{G}\boldsymbol{p}_d$. We assume that the end-effector segment of the robot, segment N, knows the target position ${}^{G}\boldsymbol{p}_d$ and is equipped with a localization sensor that can measure ${}^{G}\boldsymbol{p}_{tip}$, the 3D position of the tip of the segment in the global frame. The other segments do not have information about ${}^{G}\boldsymbol{p}_d$. Segment N can be considered a "leader" agent in that it knows ${}^{G}\boldsymbol{p}_d$ and moves toward this position, whereas the other segments are "follower" agents that reconfigure themselves in a coordinated fashion such that the tip of the robot is regulated to ${}^{G}\boldsymbol{p}_d$.

The configuration of each rigid segment can be characterized by the joint variables that describe its linear and angular displacements, which are denoted by $p_i \in \mathbb{R}$ and $\theta_i \in \mathbb{R}$, respectively. Given the joint variables, the local position vector of the prismatic joint is written as ${}^{i}\boldsymbol{p}_i = p_i {}^{i}\boldsymbol{e}_i$, where ${}^{i}\boldsymbol{e}_i \in \mathbb{R}^3$ is the unit vector along the prismatic joint expressed in the segment's local frame. The angular velocity of the segment in the global frame can be written as $\boldsymbol{\omega}_i = [0 \;\; 0 \;\; \dot{\theta}_i]^T \in \mathbb{R}^3$. We denote the vector of generalized coordinates of the i-th segment as $\boldsymbol{q}_i \in \mathbb{R}^6$, which is given by:

$$ \boldsymbol{q}_i = [\; {}^{i}\boldsymbol{p}_i^T \;\; 0 \;\; 0 \;\; \theta_i \;]^T . \tag{2} $$

The position regulation controller is defined as:

$$ \dot{\boldsymbol{q}}_i = \boldsymbol{G}_i^{\dagger} \left(- \sum_{j \in \mathcal{N}_i} ({}^{i}\boldsymbol{p}_i - {}^{i}\boldsymbol{R}_j {}^{j}\boldsymbol{p}_j) + \boldsymbol{B}_i \boldsymbol{E} \right). \tag{3} $$

The components of the controller are defined as follows. The matrix $\boldsymbol{G}_i^{\dagger}$ denotes the Moore-Penrose inverse of $\boldsymbol{G}_i \in \mathbb{R}^{3\times 6}$, which is given by:

$$ \boldsymbol{G}_i = [\; \boldsymbol{I}_{3\times 3} \;\;\; -{}^{i}\hat{\boldsymbol{p}}_i \;], \tag{4} $$

where ${}^{i}\hat{\boldsymbol{p}}_i \in \mathbb{R}^{3\times 3}$ is the skew-symmetric matrix representation of ${}^{i}\boldsymbol{p}_i$. Defining the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertex set $\mathcal{V} = \{1, ..., N\}$ contains the segment identities and the edge set $\mathcal{E}$ contains pairs of segments that can communicate with each other, also referred to as *neighboring* segments, the set $\mathcal{N}_i$ is defined as the neighbors of the i-th segment. The matrix ${}^{i}\boldsymbol{R}_j \in \mathbb{R}^{3\times 3}$ is the rotation matrix from the local coordinate system of segment j to that of segment i. The matrix $\boldsymbol{B}_i \in \mathbb{R}^{3\times 3}$ is defined as:

$$ \boldsymbol{B}_i = \begin{cases} \boldsymbol{0}_{3\times 3}, & i = 1, 2, \ldots, N-1, \\ -{}^{G}\boldsymbol{R}_N^{-1} K, & i = N, \end{cases} \tag{5} $$

where K is a tunable positive gain and ${}^{G}\boldsymbol{R}_N$ is the rotation matrix from the local coordinate system of segment N to the global frame. The vector $\boldsymbol{E} \in \mathbb{R}^3$ is defined as the error between the position of the tip of the robot and its target position, both in the global frame:

$$ \boldsymbol{E} = {}^{G}\boldsymbol{p}_{tip} - {}^{G}\boldsymbol{p}_d. \tag{6} $$

Remark III.1. *The controller in Eq. (3) is completely decentralized, in the sense that each segment only requires*

measurements of its own configuration and the configurations of its neighboring segments. Note that the controller requires only segment N to measure a position error in the global frame (the error vector $\boldsymbol{E}$).

B. Trajectory Tracking Controller

We define a reference trajectory for the robot's tip in the global frame and discretize it into a set of m points. Adopting a switching strategy, we use the decentralized position regulator in Eq. (3) to drive the tip of the robot to a position within a radius $\gamma \in \mathbb{R}_{>0}$ of each point, which represents the acceptable tracking error.

The set of target points along the reference trajectory is denoted by $\mathcal{P}$ and is defined as

$$\mathcal{P} = \left\{ {}^G\boldsymbol{p}_d^{(1)}, {}^G\boldsymbol{p}_d^{(2)}, \ldots, {}^G\boldsymbol{p}_d^{(m)} \right\}, \tag{7}$$

where ${}^G\boldsymbol{p}_d^{(1)}$ is the position of the start point on the trajectory, and ${}^G\boldsymbol{p}_d^{(m)}$ is the position of the end point. The other elements of $\mathcal{P}$ are intermediate points along the trajectory. A γ-neighborhood of point ${}^G\boldsymbol{p}_d^{(l)}$ is defined as a ball of radius γ centered at this point and is denoted by $\mathcal{B}_\gamma^{(l)}$. The controller first drives the position of the robot's tip toward the start point ${}^G\boldsymbol{p}_d^{(1)}$. Once the tip enters the ball $\mathcal{B}_\gamma^{(1)}$, the controller redefines the target point as the second point, ${}^G\boldsymbol{p}_d^{(2)}$, and drives the tip toward this point until it enters the ball $\mathcal{B}_\gamma^{(2)}$. This procedure is repeated for each successive point in $\mathcal{P}$ until the robot's tip enters the γ-neighborhood of ${}^G\boldsymbol{p}_d^{(m)}$. This switching control strategy can be written as:

$$\dot{\boldsymbol{q}}_i = \begin{cases} \boldsymbol{G}_i^\dagger \left(-\sum_{j\in\mathcal{N}_i} ({}^i\boldsymbol{p}_i - {}^i\boldsymbol{R}_j{}^j\boldsymbol{p}_j) + \boldsymbol{B}_i\boldsymbol{E}^{(l)} \right), & {}^G\boldsymbol{p}_{\text{tip}} \notin \mathcal{B}_\gamma^{(l)} \\ \boldsymbol{G}_i^\dagger \left(-\sum_{j\in\mathcal{N}_i} ({}^i\boldsymbol{p}_i - {}^i\boldsymbol{R}_j{}^j\boldsymbol{p}_j) + \boldsymbol{B}_i\boldsymbol{E}^{(l+1)} \right), & {}^G\boldsymbol{p}_{\text{tip}} \in \mathcal{B}_\gamma^{(l)} \end{cases} \tag{8}$$

where $l \in \{1, 2. \ldots, m\}$, and $\boldsymbol{E}^{(l)}$ is the vector of the error between the robot's tip and the l-th point on the reference trajectory:

$$\boldsymbol{E}^{(l)} = {}^G\boldsymbol{p}_{tip} - {}^G\boldsymbol{p}_d^{(l)}. \tag{9}$$

Note that from Eq. (5), segment N is the only segment that requires $\boldsymbol{E}^{(l)}$ in its controller, and the term $\boldsymbol{B}_i\boldsymbol{E}^{(l)}$ is the zero vector for the other segments.

IV. Stability and Convergence Analysis

In this section, we analyze the motion of the robot with the proposed controller. We first study the reconfiguration of each individual segment, and then investigate the stability and convergence properties of the closed-loop system that describes the kinematics of the robot's tip.

Lemma IV.1. *Let ${}^G\boldsymbol{p}_i$ denote the local position vector ${}^i\boldsymbol{p}_i$ of segment i in the global frame G. The decentralized control law in Eq. (3) establishes a consensus protocol for the vectors ${}^G\boldsymbol{p}_i$, with an input term defined by the error $\boldsymbol{E}$ between the robot's tip and the target position.*

Proof. The vector ${}^G\boldsymbol{p}_i$ can be written as

$${}^G\boldsymbol{p}_i = {}^G\boldsymbol{R}_i{}^i\boldsymbol{p}_i, \tag{10}$$

and its time derivative is given by

$${}^G\dot{\boldsymbol{p}}_i = {}^G\boldsymbol{R}_i{}^i\dot{\boldsymbol{p}}_i + {}^G\dot{\boldsymbol{R}}_i{}^i\boldsymbol{p}_i. \tag{11}$$

The time derivative of the rotation matrix is ${}^G\dot{\boldsymbol{R}}_i = {}^G\boldsymbol{R}_i{}^i\hat{\boldsymbol{\omega}}_i$, where ${}^i\hat{\boldsymbol{\omega}}_i \in SO(3)$ is the skew-symmetric matrix representation of the i-th segment's angular velocity expressed in its local coordinate system [32]. Hence, the second term in the right-hand side of Eq. (11) can be written as ${}^G\dot{\boldsymbol{R}}_i{}^i\boldsymbol{p}_i = {}^G\boldsymbol{R}_i{}^i\hat{\boldsymbol{\omega}}_i{}^i\boldsymbol{p}_i$. Moreover, we know that

$${}^i\hat{\boldsymbol{\omega}}_i{}^i\boldsymbol{p}_i = {}^i\boldsymbol{\omega}_i \times {}^i\boldsymbol{p}_i = -{}^i\boldsymbol{p}_i \times {}^i\boldsymbol{\omega}_i = -{}^i\hat{\boldsymbol{p}}_i{}^i\boldsymbol{\omega}_i. \tag{12}$$

Thus, Eq. (11) can be rewritten as

$${}^G\dot{\boldsymbol{p}}_i = {}^G\boldsymbol{R}_i \left({}^i\dot{\boldsymbol{p}}_i - {}^i\hat{\boldsymbol{p}}_i{}^i\boldsymbol{\omega}_i \right), \tag{13}$$

and from Eq. (2) and Eq. (4), it can be simplified to

$${}^G\dot{\boldsymbol{p}}_i = {}^G\boldsymbol{R}_i\boldsymbol{G}_i\dot{\boldsymbol{q}}_i. \tag{14}$$

Furthermore, substituting the control law in Eq. (3) for $\dot{\boldsymbol{q}}_i$, we obtain

$${}^G\dot{\boldsymbol{p}}_i = {}^G\boldsymbol{R}_i\boldsymbol{G}_i\boldsymbol{G}_i^\dagger \left(-\sum_{j\in\mathcal{N}_i} ({}^i\boldsymbol{p}_i - {}^i\boldsymbol{R}_j{}^j\boldsymbol{p}_j) + \boldsymbol{B}_i\boldsymbol{E} \right). \tag{15}$$

Using the identity $\boldsymbol{G}_i\boldsymbol{G}_i^\dagger = \boldsymbol{I}$ and the fact that ${}^G\boldsymbol{R}_i{}^i\boldsymbol{p}_i = {}^G\boldsymbol{p}_i$, Eq. (15) is reduced to

$${}^G\dot{\boldsymbol{p}}_i = -\sum_{j\in\mathcal{N}_i} ({}^G\boldsymbol{p}_i - {}^G\boldsymbol{p}_j) + {}^G\boldsymbol{R}_i\boldsymbol{B}_i\boldsymbol{E}. \tag{16}$$

Equation (16) is in the form of a consensus protocol on ${}^G\boldsymbol{p}_i$ with an input term ${}^G\boldsymbol{R}_i\boldsymbol{B}_i\boldsymbol{E}$ [33]. If we also define $\boldsymbol{P} \in \mathbb{R}^{3N}$ and $\boldsymbol{B} \in \mathbb{R}^{3N\times 3}$ as

$$\boldsymbol{P} = [{}^G\boldsymbol{p}_1^T \ {}^G\boldsymbol{p}_2^T \ \ldots \ {}^G\boldsymbol{p}_N^T]^T, \tag{17}$$

$$\boldsymbol{B} = [({}^G\boldsymbol{R}_1\boldsymbol{B}_1)^T \ ({}^G\boldsymbol{R}_2\boldsymbol{B}_2)^T \ \ldots \ ({}^G\boldsymbol{R}_N\boldsymbol{B}_N)^T]^T, \tag{18}$$

we can write the concatenated representation of Eq. (16) for all segments as

$$\dot{\boldsymbol{P}} = -\boldsymbol{\mathcal{L}}\boldsymbol{P} + \boldsymbol{B}\boldsymbol{E}, \tag{19}$$

where $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{3N\times 3N}$ is defined as

$$\boldsymbol{\mathcal{L}} = \boldsymbol{L} \otimes \boldsymbol{I}_{3\times 3}, \tag{20}$$

in which $\boldsymbol{L} \in \mathbb{R}^{N\times N}$ is the Laplacian matrix of the graph associated with the communication network of the robot's segments, illustrated in Fig. 2b, and $\otimes$ represents the Kronecker product. Hence, Eq. (19) is a linear consensus system [34] driven by the signal $\boldsymbol{E}$. □

The next theorem characterizes the stability of the closed-loop system.

Theorem IV.2. *The decentralized control law in Eq. (3) ensures that the position of the robot's tip, ${}^G\boldsymbol{p}_{tip}$, is globally exponentially stable to the target position, ${}^G\boldsymbol{p}_d$. Moreover, the magnitudes of all the position vectors ${}^G\boldsymbol{p}_i$ converge to a common value, and the directions of these vectors converge to the direction of ${}^G\boldsymbol{p}_d$.*

Proof. The position of the robot's tip in the global frame can be written as the vector sum of all position vectors ${}^G\boldsymbol{p}_i$ (see Fig. 3):

$$ {}^G\boldsymbol{p}_{tip} = \sum_{i=1}^{N} {}^G\boldsymbol{p}_i. \tag{21} $$

Taking the time derivative of this equation and following the same procedure that was used to obtain Eq. (10)–(16), the velocity of the robot's tip is derived as:

$$ {}^G\dot{\boldsymbol{p}}_{tip} = \sum_{i=1}^{N} \left(- \sum_{j \in \mathcal{N}_i} ({}^G\boldsymbol{p}_i - {}^G\boldsymbol{p}_j) \right) + \sum_{i=1}^{N} {}^G\boldsymbol{R}_i \boldsymbol{B}_i \boldsymbol{E}. \tag{22} $$

We can confirm that the double summation on the right-hand side of Eq. (22) is equal to the sum of the rows of the product $-\boldsymbol{\mathcal{L}}\boldsymbol{P}$. This, in turn, can be written as the product of the row sum of $-\boldsymbol{\mathcal{L}}$ and the matrix $\boldsymbol{P}$. We know that the sum of the rows of a Laplacian matrix is a zero row vector [33]. Invoking Lemma IV.1 and considering Eq. (20), we can conclude that the sum of the rows of $\boldsymbol{\mathcal{L}}$ is a zero row vector, and consequently, the double summation in Eq. (22) is zero. Also, from the definition of the matrix $\boldsymbol{B}_i$ in Eq. (5), the second summation is equal to $\boldsymbol{I}$. Therefore, Eq. (22) is reduced to

$$ {}^G\dot{\boldsymbol{p}}_{tip} = -K\boldsymbol{E}. \tag{23} $$

Using the fact that the target position is fixed, *i.e.* ${}^G\dot{\boldsymbol{p}}_d = \boldsymbol{0}$, Eq. (23) can be rewritten as

$$ \dot{\boldsymbol{E}} + K\boldsymbol{E} = \boldsymbol{0}, \tag{24} $$

which is globally exponentially stable to the equilibrium, *i.e.* $\boldsymbol{E} = \boldsymbol{0}$, for any positive K.

Furthermore, the system in Eq. (19) is a linear time-invariant system and can therefore be solved for $\boldsymbol{P}$ as follows [35]:

$$ \boldsymbol{P}(t) = \boldsymbol{P}_0 e^{-\boldsymbol{\mathcal{L}}t} + \int_0^t e^{-\boldsymbol{\mathcal{L}}(t-\tau)} \boldsymbol{B}\boldsymbol{E}(\tau) d\tau, \tag{25} $$

where $\boldsymbol{P}_0$ is the matrix $\boldsymbol{P}$ at the initial time, $t = 0$. We can also solve Eq. (24) for $\boldsymbol{E}$ as

$$ \boldsymbol{E}(t) = \boldsymbol{E}_0 e^{-Kt}, \tag{26} $$

where $\boldsymbol{E}_0$ is the initial error at time $t = 0$. Substituting the solution for $\boldsymbol{E}(t)$ from Eq. (26) into Eq. (25), and using spectral factorization of the matrix $\boldsymbol{\mathcal{L}}$ as in [36], Eq. (25) can be rewritten as:

$$ \begin{aligned} \boldsymbol{P}(t) = &\sum_{j=1}^{3N} e^{-\lambda_j t} (\boldsymbol{u}_j^T \boldsymbol{P}_0) \boldsymbol{u}_j \\ &+ \int_0^t \left(\sum_{j=1}^{3N} e^{-\lambda_j (t-\tau)} (\boldsymbol{u}_j^T \boldsymbol{B}\boldsymbol{E}_0 e^{-K\tau}) \boldsymbol{u}_j \right) d\tau, \end{aligned} \tag{27} $$

where $\lambda_j \in \mathbb{R}_{\geq 0}$ for $j \in \{1, 2, \ldots, 3N\}$ are the eigenvalues of $\boldsymbol{\mathcal{L}}$ ordered from smallest (λ_1) to largest (λ_{3N}), and $\boldsymbol{u}_j \in \mathbb{R}^{3N}$ are their corresponding normalized eigenvectors. Calculating the integral term, we obtain the following expression[1]

$$ \begin{aligned} \boldsymbol{P}(t) = &\sum_{j=1}^{3N} e^{-\lambda_j t} \left(\boldsymbol{u}_j^T \boldsymbol{P}_0 \right) \boldsymbol{u}_j \\ &+ \sum_{j=1}^{3N} \frac{\left(e^{-Kt} - e^{-\lambda_j t} \right)}{\lambda_j - K} \left(\boldsymbol{u}_j^T \boldsymbol{B}\boldsymbol{E}_0 \right) \boldsymbol{u}_j. \end{aligned} \tag{28} $$

We know that the first eigenvalue of the Laplacian matrix $\boldsymbol{L}$ is $\lambda_1(\boldsymbol{L}) = 0$, and the other eigenvalues are strictly positive [33]. Consequently, the first three eigenvalues of $\boldsymbol{\mathcal{L}}$ are $\lambda_1(\boldsymbol{\mathcal{L}}), \lambda_2(\boldsymbol{\mathcal{L}}), \lambda_3(\boldsymbol{\mathcal{L}}) = 0$, and the $3N - 3$ remaining eigenvalues are strictly positive. Thus, we can write

$$ \lim_{t \to \infty} \boldsymbol{P}(t) = \sum_{j=1}^{3} \left(\boldsymbol{u}_j^T \boldsymbol{P}_0 \right) \boldsymbol{u}_j + \frac{1}{K} \sum_{j=1}^{3} \left(\boldsymbol{u}_j^T \boldsymbol{B}\boldsymbol{E}_0 \right) \boldsymbol{u}_j, \tag{29} $$

since the exponential terms associated with the positive eigenvalues converge to zero as $t \to \infty$. Also, we can confirm that the three normalized eigenvectors associated with the three zero eigenvalues of $\boldsymbol{\mathcal{L}}$ are:

$$ \begin{aligned} \boldsymbol{u}_1 &= \frac{1}{\sqrt{N}} [1\ 0\ 0\ 1\ 0\ 0\ \ldots\ 1\ 0\ 0]^T, \\ \boldsymbol{u}_2 &= \frac{1}{\sqrt{N}} [0\ 1\ 0\ 0\ 1\ 0\ \ldots\ 0\ 1\ 0]^T, \\ \boldsymbol{u}_3 &= \frac{1}{\sqrt{N}} [0\ 0\ 1\ 0\ 0\ 1\ \ldots\ 0\ 0\ 1]^T. \end{aligned} \tag{30} $$

Substituting these vectors into Eq. (29), and using the definitions in Eq. (17) and Eq. (18), Eq. (29) can be rewritten as

$$ \lim_{t \to \infty} \boldsymbol{P}(t) = \boldsymbol{1}_N \otimes \left(\frac{1}{N} \sum_{i=1}^{N} {}^G\boldsymbol{p}_i(0) \right) + \boldsymbol{1}_N \otimes \left(-\frac{1}{N} \boldsymbol{E}_0 \right), \tag{31} $$

where $\boldsymbol{1}_N \in \mathbb{R}^N$ is the vector of all ones, and ${}^G\boldsymbol{p}_i(0)$ denotes the initial value of ${}^G\boldsymbol{p}_i$ at time $t = 0$. Finally, using Eq. (21) and the definition of the error, $\boldsymbol{E} = {}^G\boldsymbol{p}_{tip} - {}^G\boldsymbol{p}_d$, we obtain:

$$ \lim_{t \to \infty} \boldsymbol{P}(t) = \frac{1}{N} \left(\boldsymbol{1}_N \otimes {}^G\boldsymbol{p}_d \right), \tag{32} $$

which means that

$$ \lim_{t \to \infty} {}^G\boldsymbol{p}_i(t) = \frac{1}{N} \left({}^G\boldsymbol{p}_d \right), \qquad \forall i \in \{1, 2, \ldots, N\}. \tag{33} $$

This shows that the ${}^G\boldsymbol{p}_i$ vectors all converge to the same magnitude and direction, and this direction is that of the vector ${}^G\boldsymbol{p}_d$. □

Note that the final configuration of the robot, which is a straight line, is not singular, since the prismatic joints can still move the end-effector along that line, and the revolute joints can still rotate the segments of the robot.

[1] A special case is when K is set equal to a positive eigenvalue λ_j of $\boldsymbol{\mathcal{L}}$ ($j > 3$). Then the term $e^{-\lambda_j t} e^{(\lambda_j - K)\tau}$ in Eq. (27) equals $e^{-\lambda_j t} e^0 = e^{-\lambda_j t}$ and integrates to $\int_0^t e^{-\lambda_j t} d\tau = t e^{-\lambda_j t}$ in the second summation of Eq. (28), with no denominator $(\lambda_j - K)$ in this summation. The term $t e^{-\lambda_j t}$ converges to zero as $t \to \infty$, and thus Eq. (29) still holds true.

This section concludes with the following corollary, which characterizes the stability of the switching controller proposed for trajectory tracking. The corollary can be derived from the result that the closed-loop system described by Eq. (24) is globally exponentially stable.

Corollary IV.3. *The decentralized switching controller in Eq.* (8) *drives the tip of the robot to a γ-neighborhood of each point in the set $\mathcal{P}$, defined in Eq.* (7)*, in finite time if γ is chosen sufficiently small.*

Using the control law in Eq. (8), the trajectory tracking task is performed as m position regulation tasks, which are indexed by l and executed sequentially from $l = 1$ to $l = m$. We proved that the equation for the closed-loop system in a position regulation task is given by Eq. (24). Therefore, the closed-loop system for the trajectory tracking task behaves like a switching system, in which each subsystem is:

$$\dot{\boldsymbol{E}}^{(l)} + K\boldsymbol{E}^{(l)} = \boldsymbol{0}, \qquad l = 1, 2, \ldots, m \tag{34}$$

with $\boldsymbol{E}^{(l)}$ defined in Eq. (9). Equation (34) is linear and so can be solved for $\boldsymbol{E}^{(l)}$ as:

$$\boldsymbol{E}^{(l)}(t) = \boldsymbol{E}^{(l)}_{t_l} e^{-K(t-t_l)}, \quad \forall t \in [t_l, t_{l+1}), \tag{35}$$

where t_l is the time at which subsystem l becomes active, and $\boldsymbol{E}^{(l)}_{t_l}$ denotes the value of $\boldsymbol{E}^{(l)}(t)$ at time $t = t_l$. This solution converges exponentially to the target equilibrium $\boldsymbol{E}^{(l)} = \boldsymbol{0}$, which implies that the robot's tip will converge in finite time to a neighborhood of the desired point ${}^{G}\boldsymbol{p}^{(l)}_d$ and that the trajectories of the tip are bounded. Equation (35) also holds for the norm of the error:

$$||\boldsymbol{E}^{(l)}(t)|| = ||\boldsymbol{E}^{(l)}_{t_l}|| e^{-K(t-t_l)}, \quad \forall t \in [t_l, t_{l+1}). \tag{36}$$

The *dwell time* for the l-th subsystem to reach $\mathcal{B}^{(l)}_{\gamma}$ is defined as $T_l := t_l - t_{l-1}$ [37], which can be computed by setting $||\boldsymbol{E}^{(l)}(t)|| = \gamma$ in Eq. (36) and solving this equation for T_l:

$$T_l = -\frac{1}{K} \log\left(\frac{\gamma}{||\boldsymbol{E}^{(l)}_{t_l}||}\right), \qquad l = 1, ..., m. \tag{37}$$

This equation shows that the parameter γ and the intermediate points of the reference trajectory must be chosen such that $\gamma \leq ||\boldsymbol{E}^{(l)}_{t_l}||$ for each subsystem. Otherwise, $t_l < t_{l-1}$, which implies the stability of the system backward in time, and consequently, its instability forward in time.

V. Simulation Results and Discussion

In this section, we present and discuss the performance of the proposed controllers in simulation. We implemented the position regulation controller in Eq. (3) for the rigid-link equivalent of a 5-segment continuum robot, and we implemented the trajectory tracking controller in Eq. (8) for the rigid-link equivalent of a 15-segment continuum robot. We set $\gamma = 0.01$ cm in all simulations.

The position regulation control problem was simulated for two different values of the controller gain K. From Eq. (26), it is evident that the gain K determines the rate at which the position of the robot's tip converges to the target position. Equation (28) shows that this gain also affects the convergence rate of consensus among the ${}^{G}\boldsymbol{p}_i$ vectors. Moreover, we know that the smallest positive eigenvalue of the Laplacian matrix, $\lambda_2(\boldsymbol{L})$, determines the convergence rate of a consensus system [33]. We consider a 5-segment continuum robot for which $\lambda_2(\boldsymbol{L}) = 0.382$, and we simulate its motion under the position regulation controller for cases where $K > \lambda_2(\boldsymbol{L})$ and $K < \lambda_2(\boldsymbol{L})$.

In the first simulation of the position regulator, we set $K = 0.0382$. Figure 4 shows that the convergence rate of consensus among the lengths of the five prismatic joints

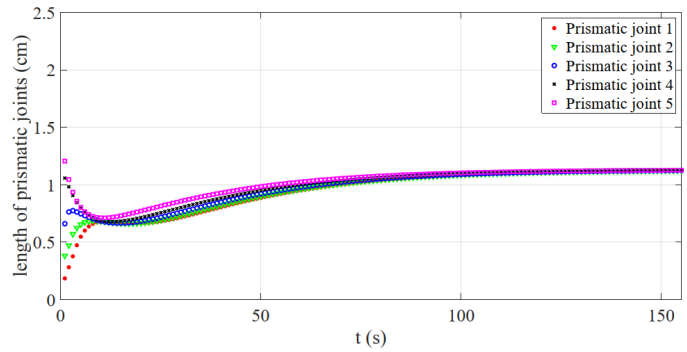

Fig. 4. Length of prismatic joints over time for a simulated 5-link serial robot, equivalent to a 5-segment soft continuum robot, that is controlled by the position regulator (3) with gain $K = 0.0382$.

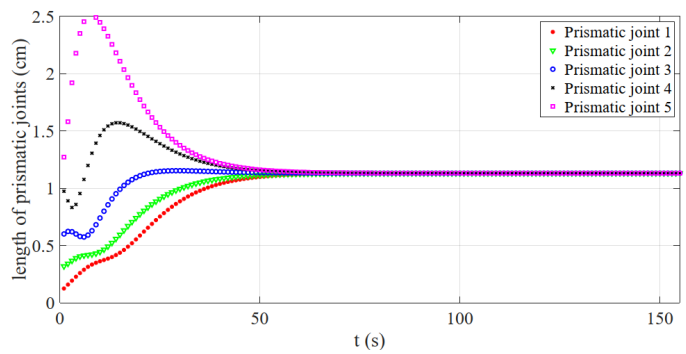

Fig. 5. Length of prismatic joints over time for a simulated 5-link serial robot, equivalent to a 5-segment soft continuum robot, that is controlled by the position regulator (3) with gain $K = 0.764$.

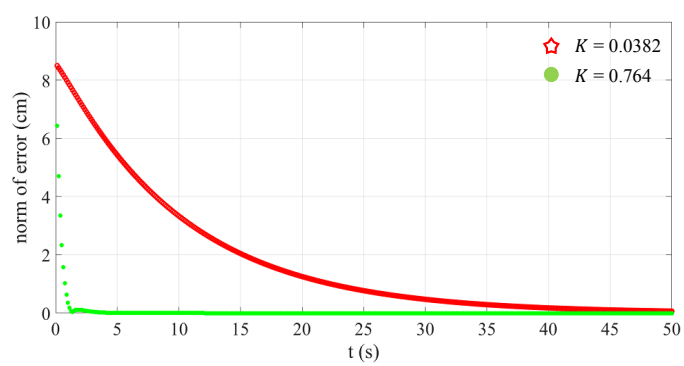

Fig. 6. Plots of tracking error $||\boldsymbol{E}(t)||_2$ over time during the position regulator simulations. The gain K = 0.764 results in faster convergence of the robot's tip to the target point than the gain K = 0.0382.

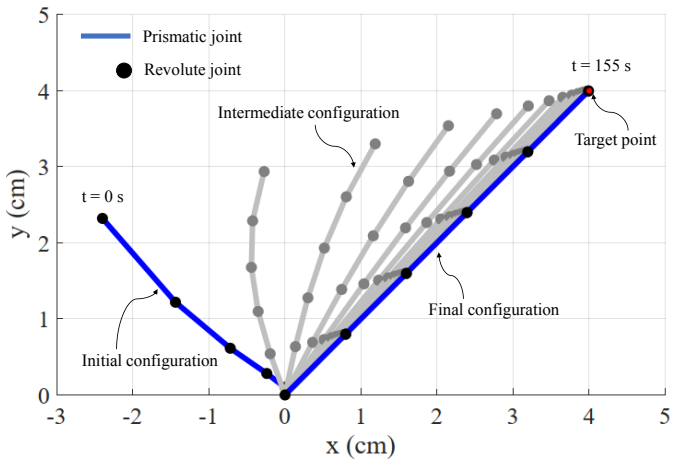

Fig. 7. Configuration over time of the simulated 5-link robot when controlled by the position regulator (3) with gain $K = 0.0382$. The robot reconfigures from its initial configuration to approach the target point with its tip. Intermediate configurations are shown in gray.

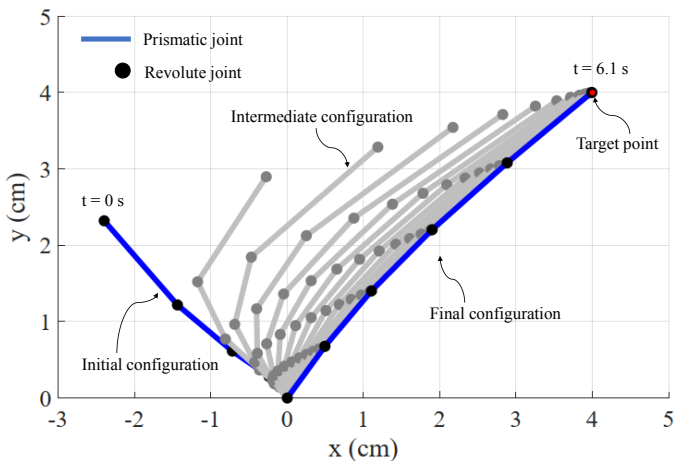

Fig. 8. Configuration over time of the simulated 5-link robot when controlled by the position regulator (3) with gain $K = 0.764$. The robot reconfigures from its initial configuration to approach the target point with its tip. Intermediate configurations are shown in gray.

to their steady-state value is relatively slow. Since $K < \lambda_2(\boldsymbol{L})$ in this case, K governs the asymptotic convergence rate of the consensus dynamics. The rate of convergence of the tracking error to zero is also determined by the value of K, and the red plot in Fig. 6 confirms that the robot's tip converges slowly to the target position. Figure 5 plots snapshots of the robot's configuration over time and illustrates that the links of the robot undergo similar changes in their lengths and orientations as the robot's tip gradually approaches the target point.

In the second simulation of the position regulator, we set $K = 0.764$. The green plot in Fig. 6 shows that the robot's tip converges very quickly to the target position. Since $\lambda_2(\boldsymbol{L}) < K$ in this case, $\lambda_2(\boldsymbol{L}) = 0.382$ governs the asymptotic convergence rate of the consensus dynamics. As Fig. 5 shows, the fact that $\lambda_2(\boldsymbol{L}) > 0.0382$ causes the prismatic joints to converge to their common steady-state value more quickly than in the first simulation, in which

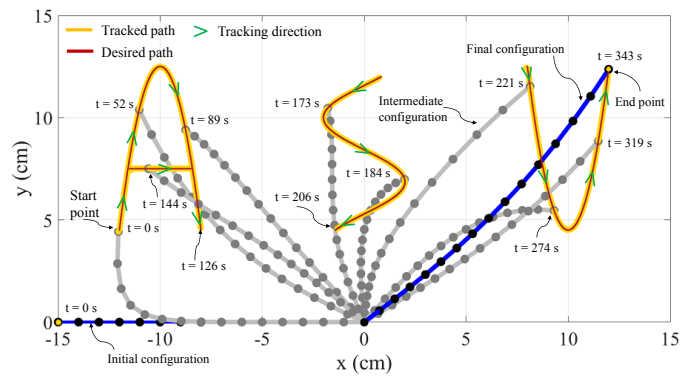

Fig. 9. Configuration over time of a simulated 15-link robot, equivalent to a 15-segment soft continuum robot, when controlled by the trajectory tracking controller (8) with gain $K = 0.25$. The robot reconfigures such that its tip moves between a series of points defined along a trajectory that spells ASU. Intermediate configurations and their corresponding times are shown in gray. The average RMSE of tracking is 0.024 cm.

$K = 0.0382$. However, since $K = 0.764$ governs the convergence rate of the tracking error and $\lambda_2(\boldsymbol{L}) < 0.764$, the tracking error in Fig. 6 converges at a faster rate than the consensus dynamics in Fig. 5. As a result, the lengths of the prismatic joints are still changing significantly, two with large overshoots (see Fig. 5), even after the robot's tip enters the γ-neighborhood of the target position. The snapshots in Fig. 8 illustrate the disparities in the links' lengths and orientations during the robot's quick reconfiguration to reach the target point. The fifth link undergoes a large elongation, which could exceed its prismatic joint limit in practice.

The simulation results for the position regulator suggest qualitative guidelines for selecting the gain K in the controller. K should be sufficiently large to drive the robot's tip quickly to the target point, but not much larger than $\lambda_2(\boldsymbol{L})$ if it is important to maintain fairly consistent changes in all link lengths and orientations throughout the robot's reconfiguration. For the trajectory tracking problem, we simulated a 15-link robot and defined the reference trajectory as the sequence of letters "ASU." Figure 9 shows snapshots of the robot's configuration over time as its tip tracks a sequence of points defined along the reference trajectory. We evaluated the trajectory-tracking performance of the robot by computing the average root-mean-square error (RMSE) between the reference trajectory and the trajectory of the robot's tip. This value is 0.024 cm, which is 0.16% of the initial length of the robot (15 cm). This very low error demonstrates the effectiveness of the control strategy.

VI. Conclusion

In this paper, we proposed a decentralized approach to kinematic control of soft segmented continuum robots that deform within a plane. The kinematics of the soft segments comply with the CC condition assumption, which enables us to model each segment as an equivalent rigid-link RPR mechanism. Decentralized controllers were defined for position regulation and trajectory tracking objectives, utilizing a consensus protocol in which adjacent segments share local

measurements of changes in the length and orientation of their equivalent rigid link. The controllers were validated with simulations of 5-link and 15-link robots, and the effect of the controller gain was investigated.

This decentralized control approach has the potential to be implemented on a variety of distributed robotic systems that are composed of multiple connected components or multiple freely-moving agents. Possible future work includes: (1) deriving similar consensus-based decentralized controllers for dynamic models of continuum robots; (2) extending the controllers to continuum robots that move in 3D space; (3) identifying bounds on the controller gain, K, to satisfy mechanical constraints such as limits on displacements of the robot's joints; and (4) incorporating secondary objectives into the controller design such as obstacle avoidance, joint torque reduction, joint limits, and increased manipulability.

REFERENCES

[1] L. Bakule, "Decentralized control: An overview," *Annual Reviews in Control*, vol. 32, no. 1, pp. 87–98, 2008.

[2] G. Antonelli, F. Arrichiello, F. Caccavale, and A. Marino, "Decentralized time-varying formation control for multi-robot systems," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1029–1043, 2014.

[3] D. M. Stipanović, G. Inalhan, R. Teo, and C. J. Tomlin, "Decentralized overlapping control of a formation of unmanned aerial vehicles," *Automatica*, vol. 40, no. 8, pp. 1285–1296, 2004.

[4] T. Umedachi and B. A. Trimmer, "Autonomous decentralized control for soft-bodied caterpillar-like modular robot exploiting large and continuum deformation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2016, pp. 292–297.

[5] T. Kano, T. Sato, R. Kobayashi, and A. Ishiguro, "Decentralized control of multi-articular snake-like robot for efficient locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1875–1880.

[6] R. Behrens, M. Poggendorf, E. Schulenburg, and N. Elkmann, "An elephant's trunk-inspired robotic arm-trajectory determination and control," in *ROBOTIK 2012; 7th German Conference on Robotics*. VDE, 2012, pp. 1–5.

[7] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[8] I. D. Walker, H. Choset, and G. S. Chirikjian, "Snake-like and continuum robots," in *Springer Handbook of Robotics*. Springer, 2016, pp. 481–498.

[9] H. Ananthanarayanan and R. Ordóñez, "Real-time inverse kinematics of (2n+1) DOF hyper-redundant manipulator arm via a combined numerical and analytical approach," *Mechanism and Machine Theory*, vol. 91, pp. 209–226, 2015.

[10] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *Journal of Intelligent and Robotic Systems*, vol. 3, no. 3, pp. 201–212, 1990.

[11] A. Salimi Lafmejani, A. Doroudchi, H. Farivarnejad, X. He, D. Aukes, M. M. Peet, H. Marvi, R. E. Fisher, and S. Berman, "Kinematic modeling and trajectory tracking control of an octopus-inspired hyper-redundant robot," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3460–3467, 2020.

[12] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.

[13] K. Xu and N. Simaan, "An investigation of the intrinsic force sensing capabilities of continuum robots," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 576–587, 2008.

[14] S. Neppalli and B. A. Jones, "Design, construction, and analysis of a continuum robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 1503–1507.

[15] S. B. Andersson, "Discrete approximations to continuous curves," in *IEEE International Conference on Robotics and Automation*. IEEE, 2006, pp. 2546–2551.

[16] C. Nho Cho, H. Jung, J. Son, and K. Gi Kim, "A modular control scheme for hyper-redundant robots," *International Journal of Advanced Robotic Systems*, vol. 12, no. 7, p. 91, 2015.

[17] K. Ishimura, M. Natori, and M. Wada, "Decentralized control of redundant manipulator based on the analogy of heat and wave equations," in *Distributed Autonomous Robotic Systems 5*. Springer, 2002, pp. 227–236.

[18] Y.-H. Liu and S. Arimoto, "Decentralized adaptive and nonadaptive position/force controllers for redundant manipulators in cooperations," *The International Journal of Robotics Research*, vol. 17, no. 3, pp. 232–247, 1998.

[19] S. Kimura, S. Tuchiya, and Y. Suzuki, "Decentralized autonomous mechanism for fault-tolerant control of a kinematically redundant manipulator," in *IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, vol. 3. IEEE, 1995, pp. 2540–2545.

[20] T. Kano, K. Sakai, K. Yasui, D. Owaki, and A. Ishiguro, "Decentralized control mechanism underlying interlimb coordination of millipedes," *Bioinspiration & Biomimetics*, vol. 12, no. 3, p. 036007, 2017.

[21] M. A. McEvoy and N. Correll, "Shape-changing materials using variable stiffness and distributed control," *Soft Robotics*, vol. 5, no. 6, pp. 737–747, 2018.

[22] G. Sartoretti, W. Paivine, Y. Shi, Y. Wu, and H. Choset, "Distributed learning of decentralized control policies for articulated mobile robots," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1109–1122, 2019.

[23] H. Mochiyama, E. Shimemura, and H. Kobayashi, "Control of serial rigid link manipulators with hyper degrees of freedom: shape control by a homogeneously decentralized scheme and its experiment," in *IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 1996, pp. 2877–2882.

[24] T. Vittor and R. Willgoss, "Motion analysis for decentralized control of N-module hyper-redundant manipulators," in *Australian Conference on Robotics and Automation*, 2005.

[25] A. Doroudchi, S. Shivakumar, R. E. Fisher, H. Marvi, D. Aukes, X. He, S. Berman, and M. M. Peet, "Decentralized control of distributed actuation in a segmented soft robot arm," in *IEEE Conference on Decision and Control*. IEEE, 2018, pp. 7002–7009.

[26] S. H. Sadati, L. Sullivan, I. D. Walker, K. Althoefer, and T. Nanayakkara, "Three-dimensional-printable thermoactive helical interface with decentralized morphological stiffness control for continuum manipulators," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2283–2290, 2018.

[27] J. Alonso-Mora, E. Montijano, M. Schwager, and D. Rus, "Distributed multi-robot formation control among obstacles: A geometric and optimization approach with consensus," in *IEEE International Conference on Robotics and Automation*. IEEE, 2016, pp. 5356–5363.

[28] S. Feng and H. Zhang, "Formation control for wheeled mobile robots based on consensus protocol," in *IEEE International Conference on Information and Automation*. IEEE, 2011, pp. 696–700.

[29] K. D. Listmann, M. V. Masalawala, and J. Adamy, "Consensus for formation control of nonholonomic mobile robots," in *IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 3886–3891.

[30] F. W. Grasso, "The octopus with two brains: how are distributed and central representations integrated in the octopus central nervous system?" *Cephalopod Cognition*, pp. 94–122, 2014.

[31] C. Della Santina, R. K. Katzschmann, A. Biechi, and D. Rus, "Dynamic control of soft robots interacting with the environment," in *IEEE International Conference on Soft Robotics*. IEEE, 2018, pp. 46–53.

[32] R. M. Murray, *A mathematical introduction to robotic manipulation*. CRC Press, 2017.

[33] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[34] R. O. Saber and R. M. Murray, "Consensus protocols for networks of dynamic agents," in *American Control Conference*, vol. 2, 2003, pp. 951–956.

[35] R. L. Williams and D. A. Lawrence, *Linear state-space control systems*. Hoboken, N.J.: John Wiley and Sons, Inc., 2007.

[36] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[37] D. Liberzon, *Switching in systems and control*. Springer Science & Business Media, 2003.