Kinematic Modeling and Trajectory Tracking Control of an Octopus-Inspired Hyper-Redundant Robot

Amir Salimi Lafmejani¹, Azadeh Doroudchi^{1,†}, Hamed Farivarnejad^{2,†}, Ximin He⁵, Daniel Aukes⁴, Matthew M. Peet², Hamidreza Marvi², Rebecca E. Fisher³, and Spring Berman²

Abstract-This paper addresses the kinematic modeling and control of hyper-redundant robots inspired by the octopus arm. We propose a discrete multi-segment model in which each segment is a 6-DoF Gough-Stewart parallel platform. Our model is novel in that it can reproduce all generic motions of an octopus arm, including elongation, shortening, bending, and particularly twisting, which is usually not included in such models, while enforcing the constant-volume property of the octopus arm. We use an approach that is inspired by the unique decentralized nervous system of the octopus arm to overcome challenges in solving the Inverse Kinematics (IK) problem, including the large number of solutions for this problem and the impracticality of numerical methods for real-time applications. We apply the pseudoinverse Jacobian method to design a kinematic controller that drives the tip of the hyper-redundant robot to track a reference trajectory. We evaluate our proposed model and controller in simulation for a variety of 3D reference trajectories: a straight line, an ellipse, a sinusoidal path, and trajectories that emulate octopus-like reaching and fetching movements. The tip of the simulated hyper-redundant robot tracks the reference trajectories with average root-mean-square errors that are less than 0.3% of the robot's initial length, demonstrating the effectiveness of our modeling and control approaches.

Index Terms—Redundant Robots, Parallel Robots, Underactuated Robots, Motion and Path Planning, Kinematics

I. INTRODUCTION

O CTOPUSES are intelligent, behaviorally complex invertebrates with eight continuously deformable arms. Due to their versatile motions, octopus arms provide a source of inspiration for the design of extensible hyper-redundant

This paper was recommended for publication by Editor Kyu-Jin Cho upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by Office of Naval Research Award N00014-17-1-2117 and by the Arizona State University Global Security Initiative.

¹A. 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.

²H. Farivarnejad, S. Berman, H. Marvi, and M. M. Peet are with the School for Engineering of Matter, Transport and Energy, ASU, Tempe, AZ 85287 {hamed.farivarnejad, spring.berman, hmarvi, mpeet}@asu.edu.

³R. E. Fisher is with the Department of Basic Medical Sciences, University of Arizona College of Medicine-Phoenix, Phoenix, AZ 85004 and the School of Life Sciences, ASU, Tempe, AZ 85287 {rfisher@email.arizona.edu}.

⁴D. Aukes is with the Polytechnic School, ASU, Mesa, AZ 85212 {daukes@asu.edu}.

⁵X. He is with the Department of Materials Science and Engineering, University of California, Los Angeles, CA 90095 {ximinhe@ucla.edu}. [†]These authors contributed equally to the paper.

Digital Object Identifier (DOI): see top of this page.

robotic manipulators [1], [2]. The high-dimensional configuration space of a hyper-redundant robot enables it to navigate and perform dexterous manipulations in unstructured environments. However, this high degree of reconfigurability poses challenges for the modeling and control of such robots. Kinematic models of hyper-redundant robots differ significantly from models of conventional rigid-link robots due to their intrinsic continuous, complex, and highly compliant deformations. Whereas rigid-link robots can change their configuration only at a finite set of locations along their structure, hyper-redundant robots can change their configuration at any location; this property must be reflected in the kinematic modeling procedure [2].

Existing kinematic modeling approaches for octopusinspired hyper-redundant robots fall into three main categories [3]: (1) Constant Curvature (CC) approaches, (2) Non-Constant Curvature (NCC) (also called Variable-Curvature (VC)) approaches, and (3) discrete approaches. In the first method, the robot is represented as a continuous curve in 3D space that overlaps the arcs of a set of circles. This method is based on the CC assumption, which permits the use of integral representations and direct continuum methods in the kinematic modeling [4]. Therefore, the transformation matrices along the hyper-redundant robot backbone can be obtained from conventional techniques such as the Denavit-Hartenberg (DH) method, homogeneous transformations, exponential coordinates (screw theory), Frenet-Serret frames, and integral representations [4]. The main disadvantage of this method is that it does not model the torsion that is contributed by each of the robot's segments. The second approach compensates for limitations of the first approach; specifically, it lacks representation singularities and does not require the simplifying CC assumption. The NCC property of octopus-inspired hyper-redundant robots is more suited to the use of modal approaches [5] for kinematic modeling of hyper-redundant mechanisms. In these approaches, a sliding frame is defined at each point along the backbone of the hyper-redundant robot, and the position and orientation of the frame are represented as functions of curvature vectors. In the third approach, which was introduced in [2], curved sections of the hyper-redundant robot are approximated as rigid links, enabling the use of existing kinematic models of conventional rigid robots [6]. Each curved section of the robot is approximated as a rigid-link equivalent Revolute-Prismatic-Revolute (RPR) mechanism, which has two pairs of revolute joints located on its base and end-effector. Each pair is connected by a prismatic joint.

Given the complexity of modeling torsional and shearing

Manuscript received: September 10, 2019; Revised December 18, 2019; Accepted January 28, 2020.

movements and limitations on the mechanical design of hyperredundant robots, most prior kinematic models of octopusinspired hyper-redundant robots do not reproduce the controlled twisting and shearing motions of octopus arms [6], [7]. A model of a multi-segment hyper-redundant robot based on the anatomy of octopus arm musculature is proposed in [8], although it does not include the effects of oblique muscles that produce torsion in the arm. A kinematic model of a hyperredundant robot backbone that reproduces torsional motions is presented in [9], which demonstrates that uncontrolled torsion due to gravity and other external forces can produce large tip-tracking errors, even for relatively small values of torsion. Moreover, the kinematic models in [5], [10] do not include the constraint that the volume of an octopus arm always remains constant, called the *isovolumetric* property [6], [11].

Existing methods for kinematic control of hyper-redundant robots can be applied to octopus-inspired hyper-redundant robots, but many of them are constrained by their computational complexity and limited to robot configurations in 1D or 2D space. One widely studied approach is the fitting al*gorithm* [12], in which the configuration of a hyper-redundant robot is fit onto a continuously-curved backbone. This method is sufficient for robots with only universal joints, but it is computationally intensive. In [13], a modular Jacobian-based control scheme is proposed to drive a hyper-redundant robot to a target backbone configuration. A modal-based approach is used in [14] for a path planning algorithm for a multi-segment hyper-redundant robot. In [15], a decentralized control approach is presented for a 1D hyper-redundant robot composed of segments with local sensing, actuation, and control, and is validated in simulation. Furthermore, learning-based methods have been proposed to tackle this control problem. For example, an online reinforcement learning algorithm is used to control an octopus-like hyper-redundant robot in [16]. This approach is implemented in simulation for a 2D model of the robot and would not be efficient in real-time applications due to its computational complexity. A more computationally efficient approach for kinematic control of hyper-redundant robots is the widely-used pseudo-inverse Jacobian method [17], [18]. This method utilizes the Moore-Penrose pseudoinverse of the Jacobian matrix of a hyper-redundant robot to compute the joint velocities with minimum magnitudes that achieve the control objective. Thus, this approach can be employed to generate local solutions to the differential kinematics for each segment of a hyper-redundant robot. Moreover, this method is compatible with conventional methods for controlling hyperredundant rigid robots [19].

In this paper, we develop a kinematic model and trajectory tracking controller for a segmented hyper-redundant robot that is inspired by the octopus arm. Our kinematic model is novel in that it reproduces all fundamental motions of an octopus arm, including twisting movements, while enforcing the isovolumetric property of the arm. We derive the kinematic model of the robot in Section II. In Section II-A, we first model each segment of the robot as a 6-DoF Gough-Stewart parallel platform [20], [21], and then model the entire multisegment robot, which can replicate the diverse motions that are generated by an octopus arm. We discuss the solutions

of the Forward Kinematics (FK) and Inverse Kinematics (IK) problems for the robot in Section II-B. In order to solve the IK problem, we employ a technique similar to the Distributed Inverse Kinematics approach in [22], which is inspired by the propagation of sensorimotor information between neighboring segments of an octopus arm. As described above, most existing strategies for control of hyper-redundant robots employ numerical methods for solving the IK problem that are too computationally demanding to implement for real-time control. Our proposed methodology for designing a position regulator and trajectory tracking controller is based on the pseudo-inverse Jacobian method, which is described in Section III. In Section IV, we validate our modeling and control approach through simulations of a segmented hyper-redundant robot that tracks five different 3D reference trajectories.

II. KINEMATIC MODEL OF HYPER-REDUNDANT ROBOT

In this section, we derive the kinematic model of an octopusinspired multi-segment hyper-redundant robot in which each segment is a 6-DoF Gough-Stewart (GS) platform [20]. We first derive the kinematic model of a single-segment 6-DoF GS platform without considering the active prismatic joints on its base and end-effector. Then, we include a model of the radial prismatic joints as kinematic constraints that maintain the isovolumetric property. As in [20], we represent the kinematics of the entire hyper-redundant robot by generalizing the kinematic equations of a single-segment 6-DoF GS platform to a multisegment hyper-redundant robot. Moreover, we describe the FK and IK problems for our proposed model of the multi-segment hyper-redundant robot.

A. Gough-Stewart Platform Model of Robot

We model each segment of the hyper-redundant robot as a 6-DoF UPS Gough-Stewart (GS) platform [20], shown in Fig. 1. This 6-DoF parallel platform consists of twelve active prismatic joints, six passive universal joints, and six passive spherical joints. Six longitudinal prismatic joints connect the universal joints on the base platform to the spherical joints on the end-effector. The radial prismatic joints at the base and the end-effector of each segment are rotated 120° with respect to the local coordinate frame, as shown in Fig. 1.

The 6-DoF GS platform in Fig. 1 can generate all possible reconfigurations of each segment of the hyper-redundant robot [23], [24]; i.e., it can produce three rotations around the segment's X_a , Y_a , and Z_a axes and three translations along these axes. Pure elongations and contractions of the segment along the Z_a axis are generated when all six longitudinal prismatic joints lengthen or shorten simultaneously by the same amount. If these lengths change by different amounts, then the segment's end-effector rotates, causing the segment to bend. As a special case, torsional motion about the segment's Z_a axis is produced when every other longitudinal prismatic joint lengthens by the same amount, while the other longitudinal prismatic joints shorten by this amount.

Since the single-segment model captures all possible rotations and translations of an individual segment, the model of the entire multi-segment robot can reproduce four fundamental



Fig. 1: Model of a single segment of the hyper-redundant robot based on a UPS 6-DoF Gough-Stewart (GS) platform.



Fig. 2: Multi-segment model of an octopus-inspired hyperredundant robot composed of a series of 6-DoF GS platforms.

motions of an octopus arm: bending, elongation, shortening, and twisting [11]. Figure 2 illustrates a model of the hyperredundant robot as a series of identical interconnected segments, each modeled as the 6-DoF GS platform shown in Fig. 1. The base platform of the *i*-th segment is the endeffector of the (i - 1)-th segment. We make the following assumptions about the capabilities of the segments: (1) Each segment can measure the 3D pose of its end-effector with respect to its local coordinate frame, which is fixed to its base. (2) Each segment can transmit these measurements to its two adjacent segments, similar to the propagation of sensorimotor information through the decentralized nervous system of an octopus arm [25]. (3) The 3D pose of the robot's tip with respect to the global coordinate frame can be measured, analogous to the use of vision feedback in the octopus [26].

Let k denote the index of the joints and i = 1, 2, ..., N be the indices of the segments, where the end-effector of the Nth segment is the distal tip of the robot. We define ${}^{G}p_{i} \in \mathbb{R}^{3}$ as the position of the end-effector of the *i*-th segment in the global coordinate frame G. The rotation matrix ${}^{G}\mathbf{R}_{i} \in \mathbb{R}^{3\times 3}$ determines the orientation of the *i*-th segment's end-effector in the global frame. The vector ${}^{G}a_{k,i} \in \mathbb{R}^{3}$ denotes the position of the k-th universal joint on the base of *i*-th segment in the global frame, and ${}^{i}b_{k,i} \in \mathbb{R}^{3}$ denotes the position of the kth spherical joint on the end-effector of the *i*-th segment in the local coordinate frame of the *i*-th segment. The vector ${}^{G}l_{k,i} \in \mathbb{R}^{3}$ is the position of the k-th longitudinal prismatic joint of the *i*-th segment in the global frame. The position vector ${}^{G}\boldsymbol{p}_{i}$ is calculated as:

$${}^{G}\boldsymbol{p}_{i} = {}^{G}\boldsymbol{a}_{k,i} + {}^{G}\boldsymbol{l}_{k,i} - {}^{G}\mathbf{R}_{i}{}^{i}\boldsymbol{b}_{k,i}. \tag{1}$$

The pose of each segment in the global frame can be determined by the corresponding position vector ${}^{G}p_{i}$ and rotation matrix ${}^{G}\mathbf{R}_{i}$. The position vector ${}^{i-1}p_{i}$ and rotation matrix ${}^{i-1}\mathbf{R}_{i}$ for the *i*-th segment in the coordinate frame attached to the end-effector of segment i - 1 are defined as:

$${}^{i-1}\boldsymbol{p}_{i} = {}^{G}\mathbf{R}_{i-1}^{-1}{}^{G}\boldsymbol{p}_{i}$$

$${}^{i-1}\mathbf{R}_{i} = {}^{G}\mathbf{R}_{i-1}^{-1}{}^{G}\mathbf{R}_{i},$$
(2)

where ${}^{G}\mathbf{R}_{i-1}^{-1}$ is the inverse of the rotation matrix ${}^{G}\mathbf{R}_{i-1}$. As in [20], by rearranging Eq. (1) to solve for ${}^{G}\boldsymbol{l}_{k,i}$, using Eq. (2) to substitute in the position vector and rotation matrix that are defined in local coordinate frames, and right-multiplying the left and right sides of the resulting equation by their respective transposes, we can obtain the lengths of the longitudinal prismatic joints, $l_{k,i}$, from the following equation:

$${}^{G}\boldsymbol{l}_{k,i}^{\mathrm{T}}{}^{G}\boldsymbol{l}_{k,i} = l_{k,i}^{2}$$

$$= ({}^{i-1}\boldsymbol{p}_{i} + {}^{i-1}\boldsymbol{b}_{k,i}^{\mathrm{T}}{}^{i-1}\boldsymbol{b}_{k,i} + {}^{i-1}\boldsymbol{a}_{k,i}^{\mathrm{T}}{}^{i-1}\boldsymbol{a}_{k,i} - 2{}^{i-1}\boldsymbol{p}_{i}^{\mathrm{T}} \quad (3)$$

$${}^{i-1}\boldsymbol{a}_{k,i} + 2{}^{i-1}\boldsymbol{p}_{i}^{\mathrm{T}i-1}\boldsymbol{b}_{k,i} - 2{}^{i-1}\boldsymbol{b}_{k,i}^{\mathrm{T}}{}^{i-1}\boldsymbol{a}_{k,i})^{1/2}.$$

Let r_{ai} and r_{bi} denote the radii of the base and end-effector, respectively, of the *i*-th segment. Let ^{*i*}**A** and ^{*i*}**B** denote matrices whose columns are the position vectors $^{i-1}a_{k,i}$ and $^{i-1}b_{k,i}$ represented in the frame attached to the end-effector of the (i-1)-th segment. These matrices are defined as:

$${}^{i}\mathbf{A} = {}^{i-1}\mathbf{B} = r_{ai} \begin{bmatrix} -\frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 & 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & -1 & -1 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$
$${}^{i}\mathbf{B} = {}^{i+1}\mathbf{A} = r_{bi} \begin{bmatrix} 0 & 0 & -\frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \\ 1 & 1 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$
(4)

We define α as the ratio between the radius of the *i*-th segment's base, r_{ai} , and the radius of its end-effector, r_{bi} :

$$\alpha = \frac{r_{bi}}{r_{ai}}, \quad \alpha \in (0 \ 1]. \tag{5}$$

In order to enforce the isovolumetric property in the kinematic model of the robot, the lengths of the radial prismatic joints are determined according to the lengths of the longitudinal prismatic joints so as to keep the volume of each segment constant. Since the total volume of the robot is the sum of all the segment volumes, the total robot volume remains constant as well. The volume of the *i*-th segment, denoted by s_i , can be computed as:

$$s_i = \frac{\pi}{3} ||^{i-1} \boldsymbol{p}_i| |r_{ia}^2 (1 + \alpha + \alpha^2).$$
 (6)

To keep the volume s_i of the *i*-th segment constant, the radius of its base should be:

$$r_{ia} = \frac{1}{\sqrt{\pi}} \left[\frac{3s_i}{||^{i-1} \boldsymbol{p}_i|| (1+\alpha+\alpha^2)} \right]^{\frac{1}{2}}.$$
 (7)

B. Forward and Inverse Kinematic Problems

Here, we discuss possible solutions to the FK and IK problems for the proposed multi-segment hyper-redundant robot. The transformation matrix of the robot's tip, i.e., the *N*-th segment's end-effector, represented in the global frame, ${}^{G}\mathbf{T}_{N} \in \mathbb{R}^{4\times 4}$, can be calculated by consecutive post-multiplication of the transformation matrix of each segment's end-effector, ${}^{i-1}\mathbf{T}_{i}$, represented in the local frame attached to the end-effector of the segment i-1:

$${}^{G}\mathbf{T}_{N} = {}^{G}\mathbf{T}_{1} \cdot {}^{1}\mathbf{T}_{2} \cdot \dots \cdot {}^{N-1}\mathbf{T}_{N}.$$
(8)

The solution of the FK problem is straightforward to compute if each segment *i* is able to measure its local transformation matrix, ${}^{i-1}\mathbf{T}_i$, as we specified in assumption (1). In other words, the pose of the robot's tip can be computed from the pose of each segment's end-effector, expressed in the segment's local coordinate frame.

The IK problem for the robot is defined as finding the configuration of the entire robot, i.e., all transformation matrices on the right-hand side of Eq. (8), given the pose of the robot's tip, ${}^{G}\mathbf{T}_{N}$. This problem has a large number of solutions due to the kinematic hyper-redundancy of the robot. Different methods of solving the IK problem for a multi-segment hyper-redundant robot are discussed in [5]. These approaches utilize computationally intensive numerical methods and can converge to physically impossible solutions. Moreover, they require simplifying assumptions and approximations that preclude the reproduction of some important features of the octopus arm. We adopt the Distributed Inverse Kinematics (DIK) approach presented in [22], [27] in order to deal with the challenge of solving the IK problem and compute the configuration of the robot, given the pose of the robot's tip. In [22], the authors introduce this approach for a multi-segment planar robot composed of segments that can autonomously change their curvature using local computation, sensing, actuation, and communication. Using this method, the solution to the IK problem is determined by utilizing the communication between the segments to share the transformation matrices and curvatures between neighboring segments of the robot.

III. KINEMATIC CONTROL OF ROBOT MOTION

In this section, we describe the calculation of the Jacobian matrix of our multi-segment hyper-redundant robot and propose a differential kinematic control approach for the robot based on the pseudo-inverse Jacobian method [18]. The objective is to design controllers for regulation (position control) and trajectory tracking problems. While we do not impose limits on the joint torques here, such constraints could be included in the robot control strategy as a second prioritized task [28].

A. Jacobian Matrix of the Robot

From assumptions (1) and (2) in Section II-A, the *i*-th segment can measure its position vector ${}^{i-1}p_i$ and rotation matrix ${}^{i-1}\mathbf{R}_i$, both represented in its local coordinate frame, and can communicate these measurements to its neighboring

segments, segments i - 1 and i + 1. We define the pose of the robot's *N*-th segment as $\mathbf{X} := [{}^{G} \boldsymbol{p}_{tip}^{T} \ \boldsymbol{\delta}^{T}]^{T} \in \mathbb{R}^{6}$, where ${}^{G} \boldsymbol{p}_{tip} \in \mathbb{R}^{3}$ is the position of the robot's tip in the global frame, and $\boldsymbol{\delta} = [\psi \ \theta \ \phi]^{T} \in \mathbb{R}^{3}$ is the vector of Euler angles representing the orientation of the *N*-th segment. We also denote the joint space of the robot by $\boldsymbol{Q} \in \mathbb{R}^{6N}$, which is defined as

$$\boldsymbol{Q} = [l_{1,1} \ \dots \ l_{6,1} \ l_{1,2} \ \dots \ l_{6,2} \ \dots \ \dots \ l_{1,N} \ \dots \ l_{6,N}]^{\mathrm{T}}.$$
 (9)

The twist of the robot's tip, \dot{X}_{tip} , is mapped to the velocities of all active prismatic joints, \dot{Q} , by the Jacobian matrix of the entire robot, $\mathbf{J}_{robot} \in \mathbb{R}^{6 \times 6N}$:

$$\dot{\boldsymbol{X}}_{\text{tip}} = \boldsymbol{J}_{\text{robot}} \dot{\boldsymbol{Q}}.$$
 (10)

The Jacobian matrix $\mathbf{J}_{\text{robot}}$ can be calculated as follows. Let $\mathbf{J}_i \in \mathbb{R}^{6 \times 6}$, i = 1, 2, ..., N, denote the Jacobian matrix and $i^{-1}s_{k,i}$ denote the unit vector that indicates the direction of the prismatic joints of the *i*-th segment in its local coordinate frame. As in [20], this matrix is computed as

$$\mathbf{J}_{i} = \begin{bmatrix} i^{-1} \boldsymbol{s}_{k,i} & i^{-1} \boldsymbol{b}_{k,i} \times i^{-1} \boldsymbol{s}_{k,i} \end{bmatrix}^{-\mathrm{T}}.$$
 (11)

We define ${}^{G}\widehat{p}_{i} \in \mathbb{R}^{3\times3}$ as the cross product matrix of the position vector ${}^{G}p_{i}$. The adjoint matrix for the *i*-th segment, $\mathbf{Ad}_{\mathbf{T}_{i}} \in \mathbb{R}^{6\times6}$, is then given by [29]

$$\mathbf{Ad}_{\mathbf{T}_{i}} = \begin{bmatrix} {}^{G}\mathbf{R}_{i} & {}^{G}\widehat{\boldsymbol{p}}_{i}{}^{G}\mathbf{R}_{i} \\ \mathbf{0}_{3\times3} & {}^{G}\mathbf{R}_{i} \end{bmatrix}.$$
 (12)

Given these matrices, the Jacobian matrix of the robot can be written as

$$\mathbf{J}_{\text{robot}} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{A}\mathbf{d}_{T_1}\mathbf{J}_2 & \dots & \mathbf{A}\mathbf{d}_{T_{N-1}}\mathbf{J}_N \end{bmatrix}.$$
(13)

As Eq. (12) shows, ${}^{G}\mathbf{R}_{i}$ and ${}^{G}p_{i}$ are required to calculate the corresponding adjoint matrix. This information is contained in the transformation matrix of each segment *i* in the global coordinate frame, defined as

$${}^{G}\mathbf{T}_{i} = \begin{bmatrix} {}^{G}\mathbf{R}_{i} & {}^{G}\boldsymbol{p}_{i} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}.$$
 (14)

From assumption (3) in Section II-A, the transformation matrix of the robot's tip in the global frame, ${}^{G}\mathbf{T}_{tip}$, is known. Since each segment *i* can communicate its measurements of ${}^{i-1}p_{i}$ and ${}^{i-1}\mathbf{R}_{i}$ to its neighboring segments, as ensured by assumption (2), the transformation matrix ${}^{i-1}\mathbf{T}_{i}$ can be computed by each segment *i*. Thus, the robot can calculate the transformation matrix of each segment in the global frame as follows:

$${}^{G}\mathbf{T}_{i-1} = {}^{G}\mathbf{T}_i \cdot {}^{i-1}\mathbf{T}_i^{-1}.$$

$$(15)$$

Therefore, the transformation matrix corresponding to each segment in the global coordinate frame can be found by first solving Eq. (15) for segment N-1, then for segment N-2, and so on until the transformation matrix of each segment from the tip to the base of the robot is computed. Consequently, we can readily calculate the Jacobian matrix of the robot in Eq. (13).

B. Position Control

Here, we design a regulator that stabilizes the position of the tip of the robot to a fixed point in the global frame G. We define the error between the position of the tip, ${}^{G}p_{tip}$, and the desired point, ${}^{G}p_{d}$, as

$$\boldsymbol{E} = {}^{\boldsymbol{G}}\boldsymbol{p}_{\rm tip} - {}^{\boldsymbol{G}}\boldsymbol{p}_{\rm d}. \tag{16}$$

We decompose $\mathbf{J}_{\text{robot}}$ into upper and lower halves, which are denoted by $\mathbf{J}_v \in \mathbb{R}^{3 \times 6N}$ and $\mathbf{J}_\omega \in \mathbb{R}^{3 \times 6N}$, respectively. The former matrix is associated with the linear velocity of the robot's tip $({}^{G}\dot{\mathbf{p}}_{\text{tip}} = \mathbf{J}_v \dot{\mathbf{Q}})$, and the latter matrix is associated with the angular velocity of the *N*-th segment ($\dot{\boldsymbol{\delta}} = \mathbf{J}_\omega \dot{\mathbf{Q}}$). Defining $K \in \mathbb{R}_{>0}$ as a positive gain and $\mathbf{J}_v^{\dagger} \in \mathbb{R}^{6N \times 3}$ as the Moore-Penrose inverse of \mathbf{J}_v , we can design the control law for the position regulation problem as:

$$\dot{\boldsymbol{Q}} = -K \mathbf{J}_v^{\dagger} \boldsymbol{E}. \tag{17}$$

Substituting the control law in Eq. (17) into the kinematics of the robot in Eq. (10), and considering the fact that ${}^{G}\dot{p}_{d} = 0$, the equation of the closed-loop system is obtained as

$$\dot{\boldsymbol{E}} = -K \mathbf{J}_v \mathbf{J}_v^{\dagger} \boldsymbol{E}. \tag{18}$$

We can confirm that the null space of \mathbf{J}_v^{\dagger} is empty. Thus, the only solution for the equation $\mathbf{J}_v^{\dagger} \boldsymbol{E} = \mathbf{0}$ is $\boldsymbol{E} = \mathbf{0}$, which shows that the origin is the only equilibrium point for Eq. (18). We can now use the identity $\mathbf{J}_v \mathbf{J}_v^{\dagger} = \mathbf{I}$ and rewrite the closed-loop system in Eq. (18) as

$$\dot{\boldsymbol{E}} + K\boldsymbol{E} = \boldsymbol{0},\tag{19}$$

which is a globally exponentially stable system for any positive K. This shows that the tip of the robot exponentially converges to the desired point from any initial position with the proposed controller. The controller gain K governs the rate of convergence of the system's trajectories to the desired position. The proposed controller drives the tip of the robot to the target point faster with larger values of K.

C. Trajectory Tracking

We use the position regulator (17) in a switching strategy that controls the robot to track a predefined reference trajectory with its tip. The reference trajectory is discretized into a set of m points. The objective is 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 the desired points is denoted by \mathcal{P} and is defined as

$$\mathcal{P} = \left\{ {}^{G}\boldsymbol{p}_{d}^{(1)}, {}^{G}\boldsymbol{p}_{d}^{(2)}, \dots, {}^{G}\boldsymbol{p}_{d}^{(m)} \right\},$$
(20)

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

$$\dot{\boldsymbol{Q}} = \begin{cases} -K \mathbf{J}_{v}^{\dagger} \left({}^{G} \boldsymbol{p}_{\text{tip}} - {}^{G} \boldsymbol{p}_{d}^{(i)} \right), & {}^{G} \boldsymbol{p}_{\text{tip}} \notin \mathcal{B}({}^{G} \boldsymbol{p}_{d}^{(i)}, \gamma) \\ \\ -K \mathbf{J}_{v}^{\dagger} \left({}^{G} \boldsymbol{p}_{\text{tip}} - {}^{G} \boldsymbol{p}_{d}^{(i+1)} \right), & {}^{G} \boldsymbol{p}_{\text{tip}} \in \mathcal{B}({}^{G} \boldsymbol{p}_{d}^{(i)}, \gamma). \end{cases}$$

$$(21)$$

Thus, the closed-loop system behaves like a switching system in which each subsystem is given by

$$\dot{\boldsymbol{E}}^{(i)} + K\boldsymbol{E}^{(i)} = \boldsymbol{0}, \qquad (22)$$

where $E^{(i)} = {}^{G}p_{tip} - {}^{G}p_{d}^{(i)}$ is the position error associated with the *i*-th subsystem. Equation (22) is linear and can be solved for $E^{(i)}$ as

$$\boldsymbol{E}^{(i)}(t) = \boldsymbol{E}^{(i)}_{t_i} e^{-K(t-t_i)}, \quad \forall t \in [t_i, t_{i+1}),$$
(23)

where t_i is the time at which subsystem *i* becomes active, and $E_{t_i}^{(i)}$ denotes the value of $E^{(i)}(t)$ at time $t = t_i$. This solution converges exponentially to the desired equilibrium ($E^{(i)} = 0$), which consequently shows that the robot's tip will converge in finite time to a neighborhood around the desired point ${}^{G}p_{d}^{(i)}$ and that the trajectories of the tip are bounded. Equation (23) also holds for the norm of the error:

$$||\boldsymbol{E}^{(i)}(t)|| = ||\boldsymbol{E}^{(i)}_{t_i}||e^{-K(t-t_i)}, \quad \forall t \in [t_i, t_{i+1}).$$
(24)

We define the *dwell time* for the *i*-th subsystem to reach $\mathcal{B}({}^{G}\boldsymbol{p}_{d}^{(i)}, \gamma)$ as $T_{i} := t_{i} - t_{i-1}$ [30]. The dwell time can be calculated by setting $||\boldsymbol{E}^{(i)}(t)||$ equal to γ in Eq. (24) and solving this equation for T_{i} :

$$T_i = -\frac{1}{K} \log\left(\frac{\gamma}{||\boldsymbol{E}_{t_i}^{(i)}||}\right), \quad i = 1, ..., m.$$
 (25)

This equation shows that the parameter γ and the intermediate points of the reference trajectory must be chosen such that $\gamma \leq ||E_{t_i}^{(i)}||$ for each subsystem. Otherwise, t_i would be smaller than t_{i-1} , which would imply the stability of the system backward in time, and consequently, its instability forward in time. As in the position control task, the controller gain K determines the rate of the error convergence to zero for each subsystem. Moreover, since $\sum_{i=1}^{N} T_i$ is equal to the task completion time, and each time T_i is inversely proportional to K, the gain K can be selected to ensure that the robot's tip tracks a desired trajectory within a specified amount of time. An analytical procedure could be developed to select a value of K that guarantees particular time response characteristics for a given desired trajectory based on the parameters N, m, α , and γ , although this is beyond the scope of this paper.

IV. SIMULATION RESULTS

We validated our kinematic model and controller through MATLAB simulations of a multi-segment hyper-redundant robot that tracks five 3D reference trajectories with the tip of its arm. The first three reference trajectories are defined as a straight line, an ellipse, and a sinusoidal. These three trajectories are given by the following three parametric equations, respectively, in terms of time $t \in [0, 1]$:

$${}^{G}\boldsymbol{p}_{\text{d,line}} = [3t - 0.5 \quad 3t + 0.5 \quad t + 6]^{T}$$
 (26)

$${}^{G}\boldsymbol{p}_{\text{d,ellipse}} = [\cos(2\pi t) \quad \sin(2\pi t) \quad 4 + \sin(2\pi t)]^{T} \quad (27)$$

$${}^{G}\boldsymbol{p}_{d,\text{sinusoid}} = [\sin(2\pi t) \quad -1 + 4t \quad 5 + e^{t}\sin(2\pi t)]^{T}$$
 (28)

We also define two reference trajectories that qualitatively reproduce the reaching and fetching motions of an octopus arm. These two trajectories are given by the following two parametric equations, respectively, for $t \in [0, 1]$:

$${}^{G}\boldsymbol{p}_{d,\text{reach}} = [3e^{t} - 5 \quad 0.5\cos(2\pi t) \quad 1 + 6t]^{T}$$
 (29)

$${}^{G}\boldsymbol{p}_{\text{d,fetch}} = \begin{bmatrix} 3e^{1-t} - 2 & 2\cos(2\pi(1-t)) & 2 + 6(1-t) \end{bmatrix}^{T}$$
 (30)

In order to track the reaching and fetching trajectories, the robot must perform a combination of elongation, contraction, bending, and twisting motions, requiring each segment to reconfigure along a combination of its six possible DoF. Thus, these two reference trajectories provide challenging test cases for validating our tracking controller.

The number of segments significantly affects the tracking performance of the simulated robot and the calculation time of the controller. In Table I, we list this calculation time on a computer with 64-bit Intel(R) Core(TM) i7-8750H CPU for different numbers of segments. Since the robot did not exhibit effective performance at position and tracking control when the number of segments was lower than 15, we chose 20 segments for the robot in our simulations. The following other parameters and conditions were applied to all simulations. We defined m = 21 as the number of target positions for the robot's tip, including the start and the end points, along each reference trajectory. We set $\gamma = 0.05$ in Eq. (25). The parameter α in Eq. (5), the ratio between the radii of each segment's base and end-effector, was initially set to 0.095. The radius of the base of the robot and the initial length of the robot were 0.25 m and 5 m, respectively. Given the isovolumetric property described in Section II-A, all segments of the robot must maintain the same volume throughout each simulation. Since the total volume of the robot was distributed equally among its segments, the length of each segment was determined by Eq. (6). As a result, the lengths of the segments increase from the base of the robot (1^{st} segment) to its tip (Nth segment), since the radii of the segments decrease from base to tip in order to maintain the same volume for all segments and to enforce the isovolumetric property described in Section II-A. We empirically set the controller gain to K = 20, which produced accurate tracking in all simulations.

Figures 3a, 4a, 5a, and 6a plot the initial and final configurations of the simulated robot and several of its intermediate configurations, shown in a paler shade, as it tracks each reference trajectory in Eqs. (26)–(30). In all cases, the robot starts in a vertical initial configuration. Each desired reference trajectory is depicted as a red dashed line, and its start and end points are labeled. As shown in Fig. 3a, the robot exhibits bending, elongation, and twisting motions along its length in

TABLE I: Calculation time of the proposed controller in milliseconds (ms) for the simulated multi-segment robot with different numbers of segments.

Number of segments	10	15	20	25	30
Calculation time (ms)	32.14	77.48	91.17	135.84	192.66

order to track the desired linear trajectory. Figure 4a shows the shortening, twisting, and bending of the robot that enables it to track the desired elliptical trajectory. Tracking the desired sinusoidal trajectory and performing reaching and fetching tasks require the robot to exhibit all four fundamental octopuslike motions, as depicted in Figs. 5a and 6a. We evaluate the performance of the robot at tracking each reference trajectory by computing the average root-mean-square error (RMSE) between the reference trajectory and the trajectory of the robot's tip, which are plotted over time in Figs. 3b, 4b, 5b, 6b, and 6c. In each figure, the colored dots indicate the sequence of 21 target positions along the reference trajectory, and the colored dashed lines show the simulated position of the robot's tip. The average RMSE value for each reference trajectory is given in the captions of Figs. 3a, 4a, 5a, and 6a. These values are at most about 0.285% of the initial length of the robot, indicating the effectiveness of the control strategy.

V. CONCLUSION

We have proposed a novel discrete approach to kinematic modeling of an octopus-inspired, multi-segment hyperredundant robot. A bio-inspired solution that exploits the flow of information between interconnected segments is used to determine the instantaneous configuration of the robot, given the tip's pose, without solving the computationally expensive IK problem for the robot. Future work will include developing a dynamic model for the robot and designing a control approach based on task priorities [19] for the robot. Furthermore, a decentralized control approach will be developed in order to enable the robot to operate with autonomous functionality. In this decentralized framework, we can employ Linear Matrix Inequality (LMI)-based optimal control approaches, such as H_2 and H_∞ methods, in order to attenuate the effects of undesired exogenous inputs, such as actuation disturbances and measurement noise, on the performance of the robot.

REFERENCES

- [1] M. D. Grissom, V. Chitrakaran, D. Dienno, M. Csencits, M. Pritts, B. Jones, W. McMahan, D. Dawson, C. Rahn, and I. Walker, "Design and experimental testing of the OctArm soft robot manipulator," in *Unmanned Systems Technology VIII*, vol. 6230. International Society for Optics and Photonics, 2006, p. 62301F.
- [2] I. D. Walker, "Continuous backbone "continuum" robot manipulators," ISRN Robotics, vol. 2013, 2013.
- [3] 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.
- [4] 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.
- [5] I. S. Godage, E. Guglielmino, D. T. Branson, G. A. Medrano-Cerda, and D. G. Caldwell, "Novel modal approach for kinematics of multisection continuum arms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 1093–1098.



Fig. 3: (a) Snapshots of the simulated hyper-redundant robot tracking the linear trajectory Eq. (26). Average tracking RMSE = 0.032 cm. (b) Time evolution of the desired position (x,y,z) and tracked position $(x_{tip},y_{tip},z_{tip})$ of the robot's tip.

- [6] R. Kang, D. T. Branson, E. Guglielmino, and D. G. Caldwell, "Dynamic modeling and control of an octopus inspired multiple continuum arm robot," *Computers & Mathematics with Applications*, vol. 64, no. 5, pp. 1004–1016, 2012.
- [7] T. Zheng, D. T. Branson, R. Kang, M. Cianchetti, E. Guglielmino, M. Follador, G. A. Medrano-Cerda, I. S. Godage, and D. G. Caldwell, "Dynamic continuum arm model for use with underwater robotic manipulators inspired by Octopus vulgaris," in *IEEE International Conference* on Robotics and Automation. IEEE, 2012, pp. 5289–5294.
- [8] T. Zheng, D. T. Branson, E. Guglielmino, and D. G. Caldwell, "A 3d dynamic model for continuum robots inspired by an octopus arm," in *IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3652–3657.
- [9] I. D. Walker, "The importance of torsion in robot backbones," *Recent Advances in Circuits, Systems, Signal Processing and Communications. World Scientific and Engineering Academy and Society*, vol. 4, pp. 25–31, 2014.
- [10] I. S. Godage, D. T. Branson, E. Guglielmino, and D. G. Caldwell, "Path planning for multisection continuum arms," in *IEEE International Conference on Mechatronics and Automation*, 2012, pp. 1208–1213.
- [11] W. M. Kier, "The musculature of coleoid cephalopod arms and tentacles," Frontiers in Cell and Developmental Biology, vol. 4, p. 10, 2016.
- [12] R. L. Hatton and H. Choset, "Generating gaits for snake robots: annealed chain fitting and keyframe wave extraction," *Autonomous Robots*, vol. 28, no. 3, pp. 271–281, 2010.
- [13] 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.
- [14] X. Yu, X. Wang, D. Meng, H. Liu, and B. Liang, "Collision free path planning for multi-section continuum manipulators based on a modal



Fig. 4: (a) Snapshots of the simulated hyper-redundant robot tracking the elliptical trajectory (27). Average tracking RMSE = 0.044 cm. (b) Time evolution of the desired position (x,y,z) and tracked position $(x_{tip},y_{tip},z_{tip})$ of the robot's tip.

method," in IEEE International Conference on CYBER Technology in Automation, Control, and Intelligent Systems, 2018, pp. 236–242.

- [15] 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.
- [16] Y. Engel, P. Szabo, and D. Volkinshtein, "Learning to control an octopus arm with Gaussian process temporal difference methods," in Advances in Neural Information Processing Systems, 2006, pp. 347–354.
- [17] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," *IEEE Journal of Robotics and Automation*, vol. 17, no. 1-19, p. 16, 2004.
- [18] 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, 2003.
- [19] S. Soylu, B. J. Buckham, and R. P. Podhorodeski, "Dexterous taskpriority based redundancy resolution for underwater manipulator systems," *Transactions of the Canadian Society for Mechanical Engineering*, vol. 31, no. 4, pp. 519–533, 2007.
- [20] A. S. Lafmejani, M. T. Masouleh, and A. Kalhor, "Trajectory tracking control of a pneumatically actuated 6-DOF Gough-Stewart parallel robot using backstepping-sliding mode controller and geometry-based quasi forward kinematic method," *Robotics and Computer-Integrated Manufacturing*, vol. 54, pp. 96–114, 2018.
- [21] S. Pedrammehr, H. Asadi, and S. Nahavandi, "A study on vibrations of hexarot-based high-G centrifugal simulators," *Robotica*, pp. 1–18, 2019.
- [22] 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.
- [23] A. Salimi Lafmejani, B. Danaei, A. Kalhor, and M. T. Masouleh, "An experimental study on control of a pneumatic 6-DoF Gough-



Fig. 5: (a) Snapshots of the simulated robot tracking the sinusoidal trajectory Eq. (28). Average tracking RMSE = 0.049 cm. (b) Time evolution of the desired position (x,y,z) and tracked position $(x_{tip},y_{tip},z_{tip})$ of the robot's tip.

Stewart robot using backstepping-sliding mode and geometry-based quasi-forward kinematic method," in 5th RSI International Conference on Robotics and Mechatronics (ICRoM). IEEE, 2017, pp. 239–245.

- [24] M. Sharifzadeh, R. Khodambashi, and D. M. Aukes, "An integrated design and simulation environment for rapid prototyping of laminate robotic mechanisms," in ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2018.
- [25] 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.
- [26] G. Levy and B. Hochner, "Embodied organization of Octopus vulgaris morphology, vision, and locomotion," *Frontiers in Physiology*, vol. 8, p. 164, 2017.
- [27] M. A. McEvoy and N. Correll, "Distributed inverse kinematics for shapechanging robotic materials," *Procedia Technology*, vol. 26, pp. 4–11, 2016.
- [28] A. Dietrich and C. Ott, "Hierarchical impedance-based tracking control of kinematically redundant robots," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 204–221, 2020.
- [29] R. M. Murray, A mathematical introduction to robotic manipulation. CRC Press, 2017.
- [30] D. Liberzon, Switching in systems and control. Springer Science & Business Media, 2003.



Fig. 6: (a) Snapshots of the robot tracking the reaching and fetching reference trajectories, Eq. (29) and Eq. (30). Average tracking RMSEs for reaching and fetching are 0.051 cm and 0.057 cm, respectively. (b), (c) Time evolution of the desired and tracked positions of the robot's tip.