Dynamic Modeling and Control of Octopus-Inspired Soft Continuum Robots with Distributed Sensing and Actuation

by

Azadeh Doroudchi

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2022 by the
Graduate Supervisory Committee:

Spring Berman, Co-Chair
Konstantinos Tsakalis, Co-Chair
Jennie Si
Hamid Marvi

ARIZONA STATE UNIVERSITY

May 2022

ABSTRACT

Soft continuum robots with the ability to bend, twist, elongate, and shorten, similar to octopus arms, have many potential applications, such as dexterous manipulation and navigation through unstructured, dynamic environments. Novel soft materials such as smart hydrogels, which change volume and other properties in response to stimuli such as temperature, pH, and chemicals, can potentially be used to construct soft robots that achieve self-regulated adaptive reconfiguration through on-demand dynamic control of local properties. However, the design of controllers for soft continuum robots is challenging due to their high-dimensional configuration space and the complexity of modeling soft actuator dynamics. To address these challenges, this dissertation presents two different model-based control approaches for robots with distributed soft actuators and sensors and validates the approaches in simulations and physical experiments. It is demonstrated that by choosing an appropriate dynamical model and designing a decentralized controller based on this model, such robots can be controlled to achieve diverse types of complex configurations.

The first approach consists of approximating the dynamics of the system, including its actuators, as a linear state-space model in order to apply optimal robust control techniques such as H_∞ state-feedback and H_∞ output-feedback methods. These techniques are designed to utilize the decentralized control structure of the robot and its distributed sensing and actuation to achieve vibration control and trajectory tracking. The approach is validated in simulation on an Euler-Bernoulli dynamic model of a hydrogel-based cantilevered robotic arm and in experiments with a hydrogel-actuated miniature 2-DOF manipulator.

The second approach is developed for soft continuum robots with dynamics that can be modeled using Cosserat rod theory. An inverse dynamics control approach is implemented on the Cosserat model of the robot for tracking configurations that include bending, torsion, shear, and extension deformations. The decentralized controller structure facilitates its implementation on robot arms composed of independently-controllable segments that have local sensing and actuation. This approach is validated on simulated 3D robot arms and on an actual silicone robot arm with distributed pneumatic actuation, for which the inverse dynamics problem is solved in simulation and the computed control outputs are applied to the robot in real-time.

DEDICATION

I dedicate this dissertation to my family and many friends.

ACKNOWLEDGMENTS

I feel blessed to be able to follow my dreams and I would like to thank God for always being there for me!

I am grateful for receiving a lot of support from my advisor, colleagues, family, and friends. First and foremost, I am thankful to my advisor, Professor Spring M. Berman, who provided a peaceful environment for doing high-quality research. She was not only an academic advisor but also an understanding friend. None of my achievements during graduate school, including this dissertation, would have been possible without her help. I also would like to acknowledge the supporting grant from the Office of Naval Research (ONR) Award N00014-17-1-2117. I would like to express my appreciation to my committee members, Prof. Konstantinos Tsakalis, Prof. Jennie Si and Prof. Hamid Marvi for their valuable guidance and comments. I am also grateful for receiving support and helpful advice from Prof. Daniel M. Aukes, Prof. Wenlong Zhang, Prof. Rebecca E. Fisher, Prof. Matthew M. Peet, and Prof. Ximin He. I would like to acknowledge my colleagues, specifically, Mohammad Sharifzadeh, Roozbeh Khodambashi, Amir Salimi Lafmejani, Hamed Farivarnejad, Zhi Qiao, Sachin Shivakumar, and Dongting Li, from whom I received assistance during my research.

In addition, I would like to thank my parents and my siblings for all their sacrifices and supports. You have been always there for me and I love you so much. Finally, I could not thank my husband enough for his full support and help during my journey. I love you my beloved Mohammad!

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

1.1 Problem Definition

This research is primarily motivated by the ongoing development of a soft continuum robot composed of a smart hydrogel material with local sensing, actuation, and control. The robot is being designed to reproduce the highly flexible bending, elongation, shortening, and twisting movements of octopus arms, in order to perform tasks such as object grasping and manipulation and underwater propulsion via crawling or swimming. This dissertation describes my research on modeling the dynamics of this soft continuum robot and its components and designing control strategies for configuration tracking, trajectory tracking, and disturbance rejection, which are validated in simulation and on experimental prototypes with individually-controllable cubical hydrogel actuators. The configuration tracking controller is also validated experimentally on a pneumatically actuated robotic arm composed of multiple silicone segments.

1.2 Background

Soft continuum robots are fabricated from soft materials (Rus and Tolley, 2015) and are designed with a continuous backbone (Walker, 2013b). Webster III and Jones (2010a) point out that due to their hyper-redundant kinematics, these robots have high dexterity (infinite degrees of freedom) and compliance, with the ability to grasp objects and perform manipulation tasks that rigid-link robots cannot. Soft continuum robots can be used in manufacturing, surgery, and other applications requiring flexible manipulators that can operate safely in close proximity to humans. They can also be used to perform unstructured

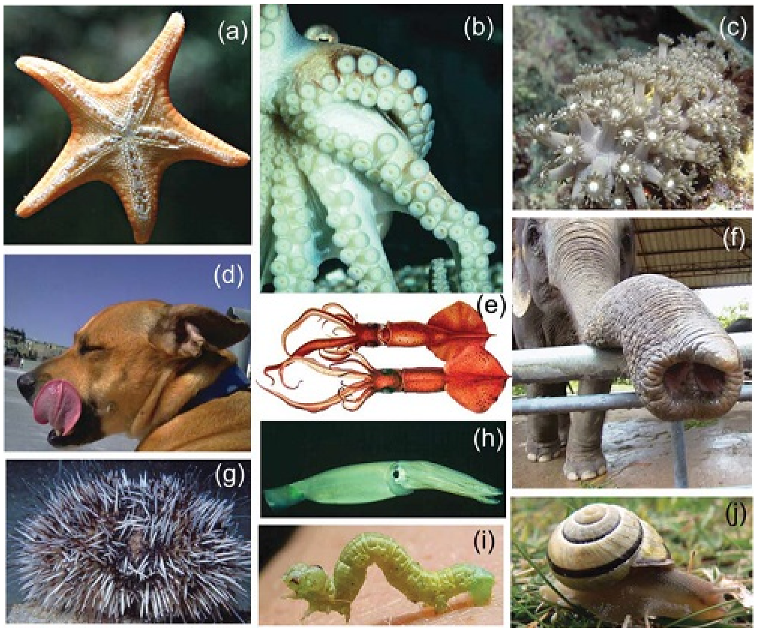


Figure 1.1: Examples of hydroskeletons and muscular hydrostats from Trivedi *et al.* (2008b): (a) tube feet in starfish, (b) octopus arms, (c) colonial anemone, (d) mammalian tongue, (e) squid, (f) elephant trunk, (g) echinoid, (h) *Illex illecebrosus*, (i) inchworm, and (j) snail foot.

manipulation and locomotion tasks in uncertain, dynamic environments.

The natural world demonstrates the potential capabilities of soft robots. There are many examples in nature of compliant, flexible structures made from soft materials, as shown in Fig. 1.1. Muscular hydrostats such as elephant trunks, mammal and lizard tongues (Kolachalama and Lakshmanan, 2020), and octopus arms are soft structures that can bend, extend, and twist. Exact mimicry of these complex structures is neither necessary nor practical for the development of soft robots. However, a fundamental understanding of the morphology and functionality of these soft structures in nature can provide insight and lead to new design concepts in soft robotics, as discussed in Trivedi *et al.* (2008b).

1.2.1 Inspiration From Octopuses

Octopuses in particular have been a rich source of inspiration for roboticists, as discussed in Cianchetti *et al.* (2011), due in part to the dexterity and flexibility of their continuously deformable arms, which are capable of four primary deformations: bending, twisting, shortening, and elongation (Kier and Smith, 1985). The octopus employs unique neuromuscular mechanisms to coordinate and control its eight arms for a diverse set of behaviors, including reaching and fetching, grasping, manipulation, walking, swimming, mimicry, squeezing through narrow passages, opening clam shells, unscrewing jar lids, and even building fortified dens. Due to this versatility, Chirikjian (1994) claims that octopus arms provide a source of inspiration for the design of soft extensible hyper-redundant robotic manipulators, also called continuum robots by Walker (2013b). The high-dimensional configuration space of a continuum 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.

The octopus arm is a muscular hydrostat comprised of three major muscle groups: longitudinal, transverse, and oblique, discussed in more detail by Kier (2016) and illustrated in Fig. 1.2 in a cross-sectional view. These muscles produce fundamental arm motions such as elongation, shortening, bending, and twisting. The octopus has the ability to elongate its arm to approximately twice its original length, as studied by Margheri *et al.* (2009). The arm elongates when its transverse muscles contract, decreasing the circumference of the arm, and the arm shortens when its longitudinal muscles contract, increasing the circumference. Bending is achieved by contracting the longitudinal muscles along one side of the arm while simultaneously contracting the transverse muscles along the opposite side of the arm. The arm is also capable of complex twisting motions, which are generated by the oblique muscles.

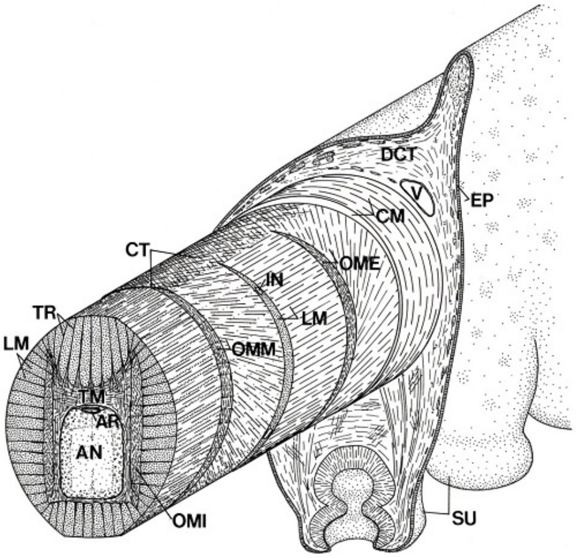


Figure 1.2: Schematic diagram of an octopus arm, showing the three-dimensional arrangement of muscle fibers and connective tissue fibers. AN, axial nerve cord; AR, artery; CM, circumferential muscle layer; CT, connective tissue; DCT, dermal connective tissue; EP, epidermis; IN, intramuscular nerve; LM, longitudinal muscle fibers; OME, external oblique muscle layer; OMI, internal oblique muscle layer; OMM, median oblique muscle layer; SU, sucker; TM, transverse muscle fibers; TR; trabeculae; V, vein. From Kier (1988).

The octopus uses combinations of these motions to perform more complicated tasks, e.g., reaching and fetching. During a reaching movement, the octopus extends one of its arms such that the tip of the arm arrives at a target object. This can be achieved by propagating a wave of muscle stiffening from the base of the arm to its tip, as discussed in Yekutieli *et al.* (2005a,b). As studied in Richter *et al.* (2015), reaching is often followed by fetching, during which the octopus retrieves a grasped object to its mouth. Reaching is accomplished mostly by simultaneous arm elongation and bending, while fetching is achieved through simultaneous arm shortening and bending.

In order to observe typical arm deformations during reaching and fetching movements

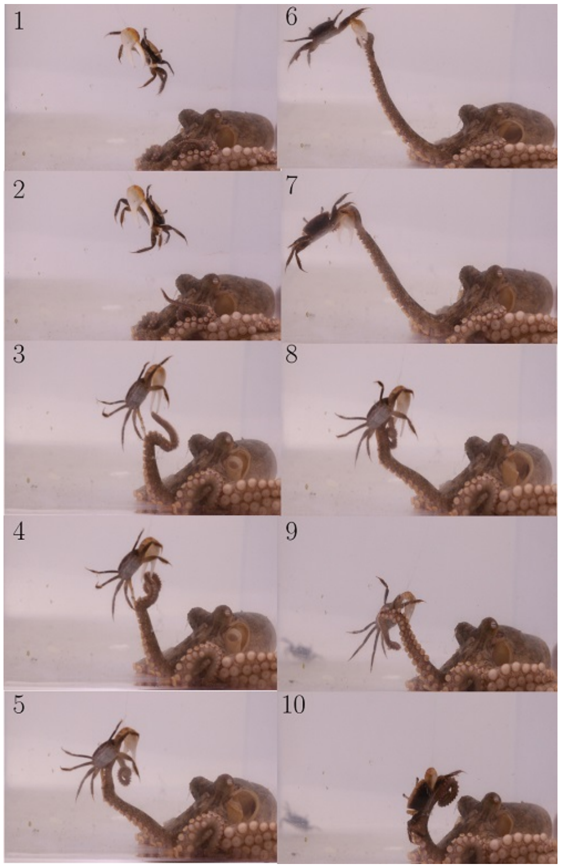


Figure 1.3: Two sequences of snapshots of a live octopus (*Octopus bimaculoides*) reaching toward (1 to 5) and fetching (6 to 10) a crab with one of its arms. (Photo credit: Hosain Bagheri, BIRTHLab, Arizona State University)

that we could then qualitatively reproduce through kinematic models and simulations, we recorded images of a live octopus (*O. bimaculoides*) that was housed in a vivarium at Arizona State University (ASU) and cared for by staff of ASU's Department of Animal Care and Technologies. Snapshots of the octopus reaching toward and fetching a live fiddler crab are shown in Fig. 1.3. Soft continuum robotic arms could be controlled to perform similar motions, for instance to direct their end-effector to a target location or to grasp and transport an object.

The octopus has a complex interconnected nervous system. In addition to the centrally

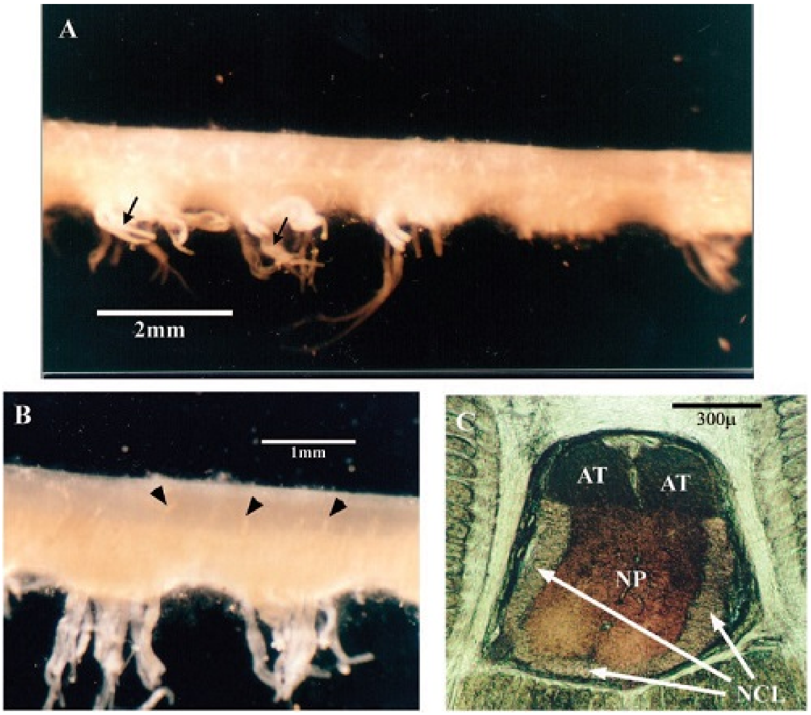


Figure 1.4: The in vitro preparation of the axial nerve cord from Gutfreund *et al.* (2006). (A) A lateral view of the isolated nerve cord (dorsal side up). Each swelling is a single brachial ganglion. The numerous ventral roots are clearly seen projecting from each ganglion (arrows). (B) Dorsal roots originate from the borderline of the axonal tract (some marked by arrowheads). (C) Unstained transverse section of the axial nerve cord at the level of a brachial ganglion, showing the two dorsally located bundles of the axonal tract (AT), the cell body or perikaryal layer (NCL), and the internal neuropil (NP) of the ganglion.

located brain, the octopus has peripheral nerves distributed through its mantle and arms that act as a decentralized network of interconnected neurons. Within each arm, there is an axial nerve cord, and these nerve cords are connected to each other via the interbrachial commissure shown in Fig. 1.4. This decentralized nervous system is organized into multiple segments along the length of the arm, as illustrated in Fig. 1.5 from Grasso (2014).

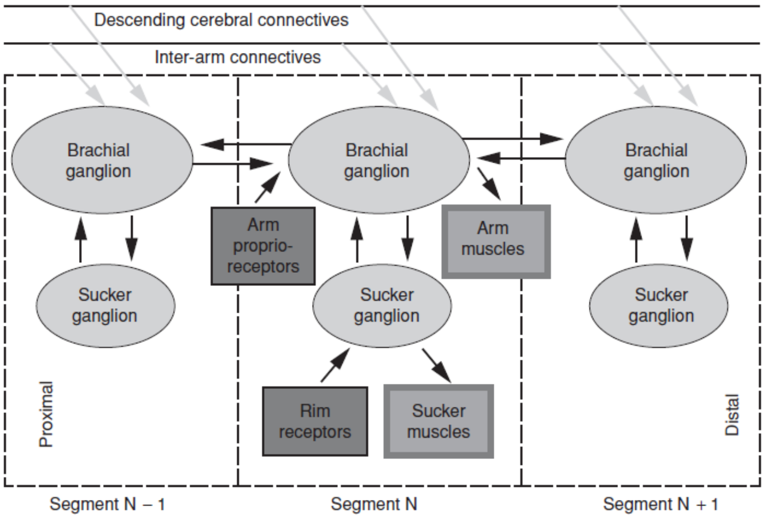


Figure 1.5: The organization of ganglia in the octopus arm, illustrated by Grasso (2014). Around each sucker is a section of the arm whose neural and muscular structure repeats along the length of the arm. The central segment, N, shows a complete set of brachial and sucker ganglia and the connections.

Chemoreceptors on the sucker rims and propioceptors embedded in the tissues of the arm provide sensory inputs to the sucker and brachial ganglia, respectively. The brachial ganglion also receives neuronal signal inputs from the sucker ganglion in its segment, descending inputs from the cerebral ganglia, and inputs from the other arms and several proximally and distally positioned brachial ganglia. The brachial ganglia process these sources of input and produce motor commands to the muscles of the arm. A shorter sensory motor loop exists from the sucker ganglia (which also processes input from the brachial ganglia in its segment) to the sucker muscles. The proximal and distal segments are included in Fig. 1.5 to indicate that this segmented organization is repeated along the length of the arm. Similar to the neuromuscular structure of an octopus arm, we consider soft robotic arms with a segmented design, in which individual segments are capable of both sensing and actuation and neighboring segments transmit information to each other. We develop dynamic models and decentralized controllers for soft robotic arms with this type of design.

1.2.2 Dynamic Modeling of Continuum Soft Robotic Arms

The kinematics and dynamics of continuum robots has been the subject of much research and has now reached a mature stage, with theoretical results now established that match most of the corresponding results for rigid-link robots. A widely-used design for soft robots that provides capabilities beyond the scope of traditional rigid-link robots is a continuous rod-like curved manipulator. A continuum robotic arm is an inherently infinite-dimensional (distributed parameter) system. Due to the segmented nature of the robot's design, the number of states or degrees of freedom of the system is limited only by the number of segments. Some examples of these soft continuum robots (Rus and Tolley, 2015), which resemble biological trunks and tentacles, are shown in Fig. 1.6. While there are many scalable and compliant soft robot designs, these designs are typically model-independent or use simplified models which do not accurately reflect either the nonlinear dynamics of highly deformable robots or the practical issues of sensor and actuator design and placement, as discussed in Kim *et al.* (2013); Laschi and Cianchetti (2014); Rus and Tolley (2015).

Dynamic models of continuum robots would facilitate a variety of control techniques. Although researchers have done some limited demonstrations of the control of soft structures, this has typically been done either through scripting or by doing kinematic rather than dynamic control. Many of the control-oriented models developed by Godage *et al.* (2015); Matsuno and Suenaga (2003); Webster III and Jones (2010b) for these types of robots have thus far been governed by kinematic equations describing rigid links and hence are not useful for designing feedback controllers when both the forces produced by the actuators and the motion of the robot are distributed throughout the structure.

A general dynamical theory of rods, which undergo large deformations in space by

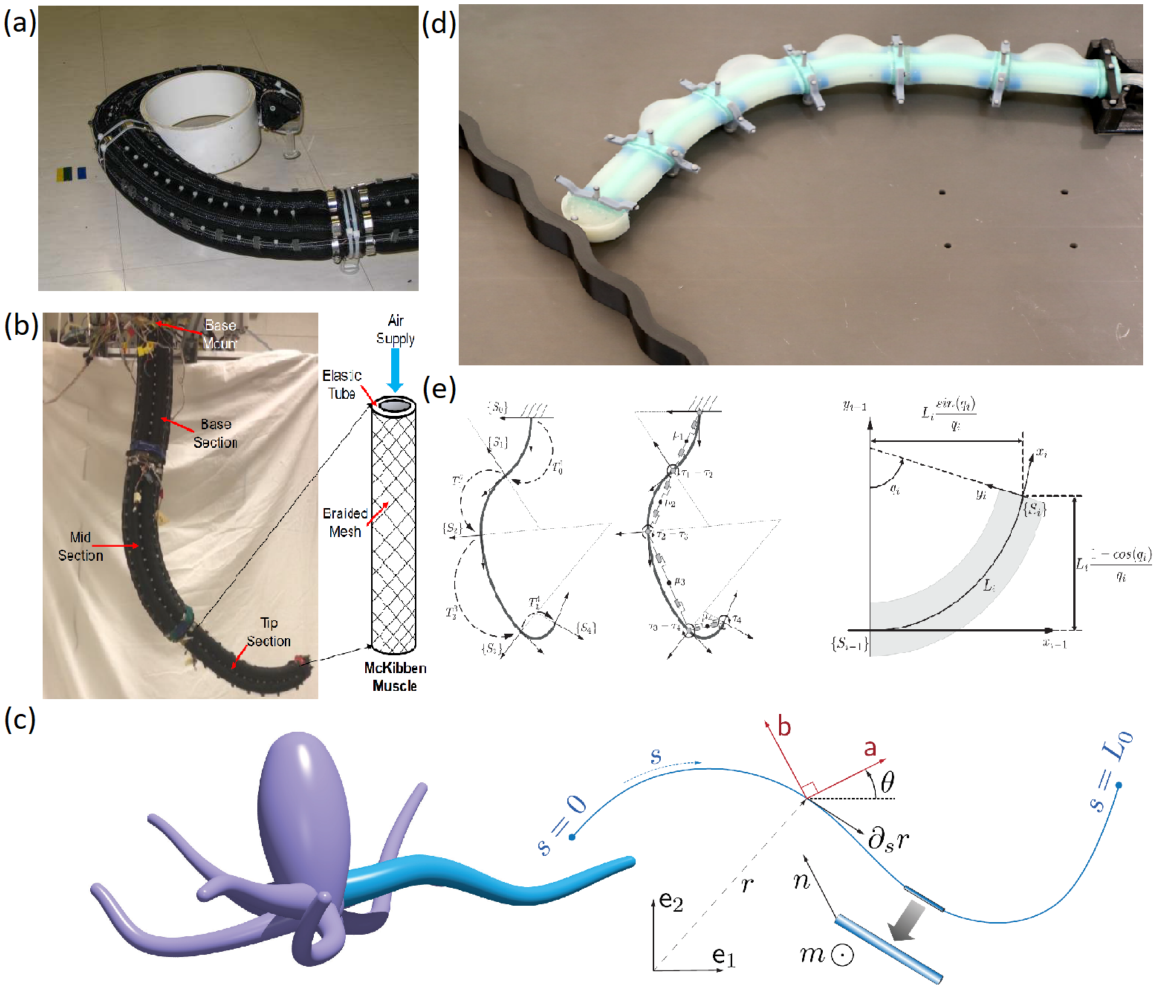


Figure 1.6: Soft continuum robots. (a) The Octarm grasping a cylindrical object (Kapadia and Walker, 2011). (b) OctArm, a pneumatically actuated continuum manipulator with three sections (Wang *et al.*, 2020). (c) Modeling an octopus arm as a special Cosserat rod (Chang *et al.*, 2020b). (d) Soft robot interaction with environment (Della Santina *et al.*, 2020). (e) The PCC (piecewise constant curvature) model of the soft robotic arm in (d).

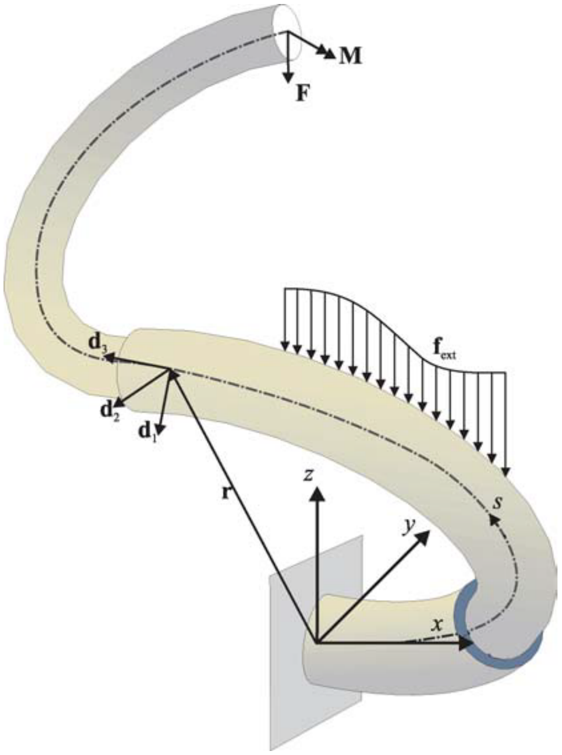


Figure 1.7: Soft robotic manipulator modeled using the Cosserat rod approach from Trivedi *et al.* (2008b), with the backbone position ($\mathbf{r}$) and orientation ($\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$) parametrized by a single variable s. The manipulator is acted upon by distributed force ($\mathbf{f_{ext}}$) and discrete forces ($\mathbf{F}$) and moments ($\mathbf{M}$).

bending, torsion, extension, and shear, can be employed for modeling continuum robots. This dynamical modeling approach is called the special Cosserat theory of rods (Antman, 1995). The geometrically exact models using Cosserat rod theory have been widely applied to static and dynamic modeling of soft continuum robots. The Kirchhoff model, a special case of the Cosserat model, can only describe bending and torsion movements; in other words, it is an unshearable and inextensible Cosserat rod model, as discussed by Till *et al.* (2019). Renda *et al.* (2016) proposed the piecewise constant strain (PCS) model, which is a continuous Cosserat model that describes shear and torsion. Cosserat theory has also been widely used in studies of different actuation systems, such as pneumatic muscle-actuated multi-section prototype continuum arms presented by Godage *et al.* (2016), fluidic actuated legs of the soft crawler FASTT presented by Renda *et al.* (2016), and tendon-driven robots by Renda and Laschi (2012). Although Godage *et al.* (2016) did not consider torsion as one of the deformations, the novelty of the study is that the model applies to variable-length multi-section continuum arms with any number of sections. Another straight-tendon robot is described by a coupled model that combines classical Cosserat rod and Cosserat string models (Rucker and Webster III, 2011), which has the advantage that it accounts for the distributed wrenches that tendons apply along the robot.

Figure 1.7 illustrates a soft robotic manipulator modeled using the Cosserat rod approach. There are examples of using Cosserat theory to model 1D elastic rods (Spillmann and Teschner, 2007) and active cannulas (Webster III *et al.*, 2008) with torsion. Lang *et al.* (2011) achieved a numerically stable and efficient method for the dynamical analysis of rods within millisecond accuracy using consistent semi-discretization of the continuous dynamical Cosserat partial differential equations (PDEs) of motion by combining two staggered spatial grids. From an algorithmic viewpoint, the parameterization of rotations by quaternions has proved to be very useful, and for time integration, standard ordinary differential equations (ODEs) and differential algebraic equations (DAEs) solvers can be applied. Till *et al.* (2015) presented an efficient numerical solution of the Cosserat kinematic model for a 6-DOF parallel continuum robot at rates that enable real-time interactive simulation, motion planning, design optimization, and control. A numerical approach for forward dynamics simulation of Cosserat-based robot models in real-time is presented by Till *et al.* (2019). The approach implicitly discretizes the time derivatives in the PDEs and then solves the resulting ODE boundary value problem (BVP) in arc length at each time step. They show that this strategy can encompass a wide variety of robot models and numerical schemes in both time and space. We have drawn on this body of work and found the Cosserat model to be applicable to the soft continuum robots that we consider.

1.2.3 Challenges of Controlling Continuum Soft Robotic Arms

Decentralized control approaches have been extensively employed in the control of distributed robotic systems such as groups of ground robots (Antonelli *et al.*, 2014) and Unmanned Aerial Vehicles (UAVs) (Stipanović *et al.*, 2004), as well as hyper-redundant soft continuum robots (Umedachi and Trimmer, 2016; Kano *et al.*, 2011; Behrens *et al.*, 2012). Decentralized control methods can be used to avoid the limitations of centralized control approaches for kinematic control (Ishimura *et al.*, 2002), cooperative control (Liu and Arimoto, 1998), and fault-tolerant control of multi-segment continuum robots (Kimura *et al.*, 1995). Moreover, decentralized control approaches can significantly reduce the computational complexity of the control strategy. The implementation of decentralized control architectures in continuum robots would enable scalability of the robot design, minimize expensive communication and power overhead, and increase robustness to partial failure, as discussed in Bakule (2008); Siljak (2011). However, challenges remain in the design of decentralized controllers for soft continuum robots.

The majority of static and dynamic controllers for continuum robots use the classic Lagrangian or Newton-Euler formulations, as introduced by Siciliano *et al.* (2010) for modeling such robots, and the Jacobian formulation for closed-loop real-time inverse dynamic control, implemented by Kapadia and Walker (2011); Godage *et al.* (2016); George Thuruthel *et al.* (2020); Wang *et al.* (2020). Using the Lagrangian formulation, Tatlicioglu *et al.* (2007) modeled and simulated the dynamics of a planar extensible continuum manipulator. Della Santina *et al.* (2020) applied a closed-loop control approach for trajectory tracking and surface following using the inverse dynamics of the Lagrangian formulation, experimentally in 2D for the first time, on a pneumatically actuated soft robotic arm. While dynamic models have been formulated, e.g. a PDE presented by Ivanescu *et al.* (2015) for a model of bending in a hyper-redundant continuum robotic arm, their complexity often prevents their practical implementation in controller design and motion planning, as noted by Walker (2013a); Walker *et al.* (2016).

Chirikjian (1994) applied a continuum approach to model the dynamics of a snake-like hyper-redundant manipulator and used this for computed torque control. Gravagne and Walker (2002) dynamically modeled the Clemson Tentacle Manipulator, a multi-section continuum robot, and showed that a PD controller plus a feed-forward regulator is sufficient for stabilizing the system. Gravagne *et al.* (2003) further developed a model that describes large deflections of the robot and applied a vibration-damping set-point controller to illustrate the efficacy of their method. Snyder and Wilson (1990) dynamically modeled polymeric pneumatic tubes subject to tip loading using a bending beam model, but did not use this for control.

A few real-time control approaches based on reduced-order finite element methods have been developed and tested on soft robots and manipulators using the Simulation Open Framework Architecture (SOFA) software framework, introduced and implemented by Duriez (2013) and Katzschmann *et al.* (2019).

To address the lack of control methodologies that take full advantage of body compliance, a compliant mechanics environment for controlling soft robots was presented by Naughton *et al.* (2020) that uses the recently developed software package Elastica, an open-source simulation environment for slender rods that can bend, twist, shear, and stretch. Other studies by Chang *et al.* (2020a,b), propose methods for controlling an octopus-inspired soft arm with muscle-like actuation. In these previous studies, only one or two of the four primary types of deformations (bending, shear, extension, and torsion) are simulated or experimentally produced in a soft continuum robot, depending on the control objective or model limitations (Boyer *et al.*, 2011; Niu *et al.*, 2019; Guan *et al.*, 2020).

We develop a *decentralized* closed-loop control approach for soft continuum robots that are described by the geometrically exact Cosserat rod model, which enables such robots to track 3D configurations that involve any combination of the four primary types of deformations. This approach involves the design of block-diagonal gain matrices that are defined in terms of the physical and material properties of distinct cross-sections of the robot arm. This structure facilitates its implementation on continuum robot arms composed of independently-controllable segments that have local sensing and actuation.

In addition, there has been significant research on the question of how to construct stabilizing decentralized feedback laws for a given network and, furthermore, whether there are necessary and sufficient conditions for the existence of such local feedback laws. Few existing soft robot designs are able to respond locally to stimuli in a scalable manner, since they use centralized information processing and the resulting feedback cannot be used to adapt to local configuration changes. The largest class of systems for which we know the answer to this question are those systems which are *quadratically invariant* (Lessard and Lall, 2011; Rotkowitz *et al.*, 2010). While testing quadratic invariance is known to be NP-hard, in practice, testing quadratic invariance under sparsity constraints for reasonably-sized systems is not difficult and furthermore, certain well-studied sparsity patterns are known to be quadratically invariant, with the most well-known case being when the controller is diagonal or both the controller and plant are upper- or lower-tridiagonal. Unfortunately, however, the tridiagonal sparsity constraint generated by discretization of beam-type equations (with zeros everywhere except the diagonal and first off-diagonal elements) is not quadratically invariant.

Because the decentralized control problem with tridiagonal structure is difficult, the literature on vibration control of beams or rods focuses on the case of diagonal decentralization, in which neighboring controllers do not communicate with each other. We were interested in designing decentralized controllers with tridiagonal structure for soft robot arms replicating the neighboring segment connections inspired by the octopus arm. Since the tridiagonal structure is not quadratically invariant, we instead considered the non-convex

Bilinear Matrix Inequality (BMI) formulation of the problem and designed algorithms to solve this BMI directly using iteration and gradient descent.

Many algorithms have been developed for finding local solutions to BMI problems, several focusing on Branch and Bound (Goh *et al.*, 1994; Tuan *et al.*, 2000; VanAntwerp and Braatz, 2000; VanAntwerp *et al.*, 1997, 1999). However, as Liu and Papavassilopoulos (1996) note, many such global optimization algorithms have high computational complexity, making them impractical for the large state spaces induced by spatial discretization of a PDE. For high-dimensional problems, Yamada and Hara (1998) suggest a modified triangle-covering based algorithm which reduces the computational cost. Unfortunately, however, this approach is restricted to a class of BMIs that does not include the decentralized controller synthesis problem. The method proposed by Kanev *et al.* (2004) and the rank minimization approach in Ibaraki and Tomizuka (2001) will both typically converge to a local optimum, given an initial feasible controller. Other approaches involve linearization of the BMI, as studied by Hassibi *et al.* (1999). Many of these methods can fail to converge to even locally optimal solutions, as shown by Safonov *et al.* (1994).

1.2.4 Novel Materials for Building and Controlling Soft Robots

Soft robot arms are usually actuated with tendon, cable, pneumatic, or hydraulic actuators, as compared by George Thuruthel *et al.* (2018). However, soft actuators are lightweight and noiseless, in contrast to pneumatic systems with pumps and motors. Robots composed of highly deformable matter such as fluids, gels, and elastomers used by Majidi (2014), with soft actuators such as shape memory alloys (SMAs) and soft sensors such as artificial skin with touch and temperature receptors, comprise a new generation of robots reviewed by Pfeifer *et al.* (2012) that are capable of flexible movements and delicate interactions. Such robots have extensive potential uses in healthcare applications, robotic exploration tasks, and cooperative human assistance. Potential applications of stimuli-responsive materials have been shown by Bacelar *et al.* (2017); Qin *et al.* (2019) in micro-manipulation, sensing, optics and in biomedical applications by Guiseppi-Elie (2010). Furthermore, continuum robots composed of soft materials would exhibit high structural compliance in response to environmental inputs that can enhance the robot's functionality. Hydrogels in particular have the ability to absorb and release water, undergoing reversible volumetric changes that facilitates their use as soft actuators, as discussed by Mishra *et al.* (2020).

Despite extensive research on macroscale robotics and micro-electromechanical systems, relatively little attention has been paid to the creation of miniature soft robots with diverse shapes, actuation mechanisms, and integrated functionalities. Robotic manipulation and locomotion in millimeter-scale robots using stimuli-responsive materials has been previously accomplished by Han *et al.* (2018); Yeghiazarian *et al.* (2005); Wang *et al.* (2015). Kwon *et al.* (2008) present miniature (micro- to millimeter) soft aquabots that combine multiple functionalities to perform multi-functional operations in aqueous environments, effectively simulating their natural counterparts. Zolfagharian *et al.* (2018) achieved substantial improvement in positioning the tip of a bending actuator made of electroactive hydrogels.

Currently, mechanisms that use stimuli-responsive materials in their structures are limited in terms of the type of motion they can produce. In addition, their motion is dictated by their geometry and material distribution (e.g., bimorph beams), which cannot be changed after manufacturing, as mentioned by Ionov (2014). This prevents these mechanisms from responding to changes in working conditions, which is a requirement in mechanisms made for exploring unstructured environments. Hines *et al.* (2017) suggest that the use of electrically responsive soft actuators is more promising, since they are compatible with conventional electronics and batteries and facilitate the integration process. McEvoy and Correll (2018) developed a soft multi-segment shape-changing robot with integrated sensing, actuation, and processor modules, and designed a distributed controller for shape control of the robot. Researchers have tried to control the motion of soft structures by embedding electronic circuits and locally stimulating material in order to create local deformations, resulting in shape change or motion of the structure. Over the past fifteen years, local actuation for soft robots has advanced, as described by Trivedi *et al.* (2008a); Cianchetti *et al.* (2011); Godage *et al.* (2016). In some studies like Yu *et al.* (2013), only the static deformations were of interest, while in others such as Wang *et al.* (2018), robot movements were achieved using open-loop, hard-coded on-off commands, making them incapable of performing precise, closed-loop controlled movements. He *et al.* (2012) present the possibility of constructing soft continuum robots using smart hydrogels with on-demand dynamic control of local properties through continuous sensing and actuation that is distributed throughout the robot. Such robots could offer new capabilities through self-regulated adaptive reconfiguration.

Novel soft materials such as smart hydrogels can change volume and other properties in response to stimuli such as temperature, pH, and chemicals, as illustrated in Fig. 1.8. Hydrogels are new materials whose properties make them excellent candidates for soft robotic and biomedical applications (Guiseppi-Elie, 2010). A hydrogel is composed of a hydrophilic multiphase polymer mixture that may exhibit properties of both solids and liquids. Its structural framework is formed from 3D networks of randomly cross-linked polymeric chains that embody three different phases: solid polymer network matrix, interstitial water or biological fluid, and ion species (Li, 2009). Hydrogels require water to function and can be used as deformable actuators that are lightweight and noiseless, in contrast to pneumatic and hydraulic actuators. There are various types of hydrogels that deform when they are stimulated with different sources of energy (Peng *et al.*, 2018; Qin *et al.*, 2019). Poly(N-isopropylacrylamide), or PNIPAAM, is a temperature-responsive hydrogel that contracts when heated by a laser, electrical resistor, hot water, or other source (Schild,

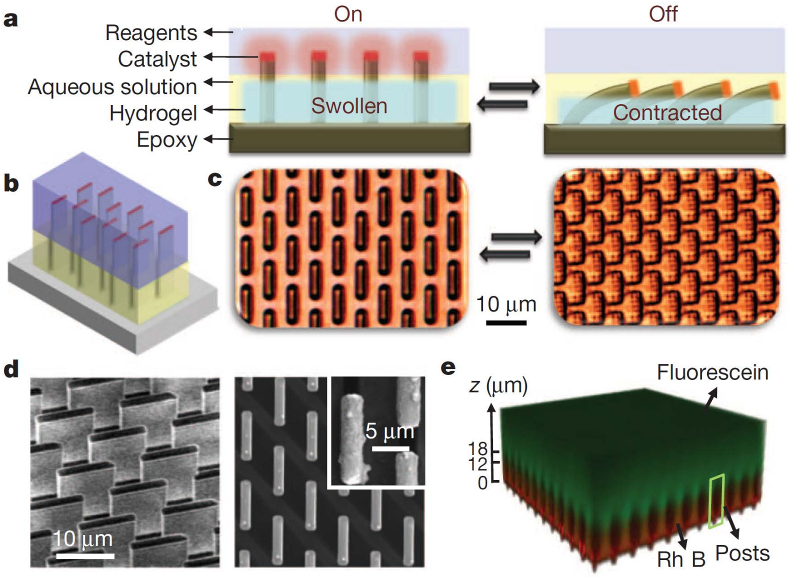


Figure 1.8: General design of SMARTS (self-regulated mechanochemical adaptively reconfigurable tunable system), from He *et al.* (2012). (a) Cross-section schematic. (b) Three-dimensional schematic. (c) Top-view microscope images of upright and bent microfins corresponding to on (left) and off (right) reaction states. (d) Forty five-degree side-view (left) and top-view (right) scanning electron microscope images of 2-mm-wide, 10-mm-long, 18-mm-high microfins with the catalyst particles on tips (inset). (e) Three-dimensional confocal microscope image of a hydrogel-embedded, 18-mm-tall post array immersed in a bilayer liquid labelled with fluorescein and rhodamine B, showing the interface height to be 12 mm.

1992). Nano-structured smart hydrogels exhibit rapid stimuli-responsive characteristics, as well as highly elastic properties that enable them to sustain large compression forces, resist slicing, and withstand deformations such as bending, twisting, and extensive stretching (Xia *et al.*, 2013).

Khodambashi *et al.* (2021) demonstrate the design of millimeter-scale hydrogel cubes with embedded Joule heaters, called soft voxel actuators (SVAs). As illustrated in Fig.

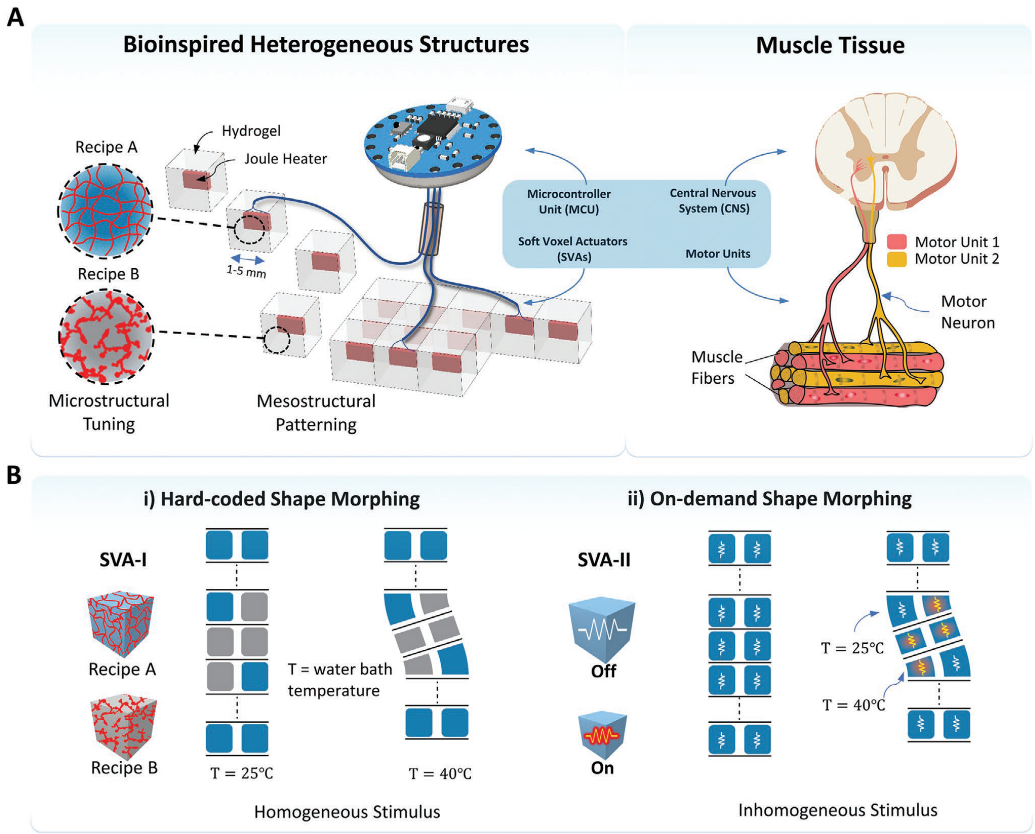


Figure 1.9: Illustration of bioinspired heterogeneous hydrogel structures composed of tunable and addressable voxels (Khodambashi *et al.*, 2021). (A) Soft voxel actuators (SVAs) are electrically addressable building blocks whose deformations can be controlled by a microcontroller unit (MCU). SVAs are analogous to motor units, consisting of a motor neuron and associated muscle fibers, which deform in response to electrical impulses from the CNS. The micro-structure of the hydrogels used to make SVAs can be altered, resulting in tunable material properties. (B) i) SVAs without embedded Joule heaters (SVA-I) are used to create structures with hard-coded shape morphing that respond to a homogeneous temperature field acting globally on the entire structure through the surrounding water bath. ii) SVAs with embedded heaters (SVA-II) are used to create structures with on-demand shape morphing by forming an inhomogeneous temperature field throughout the structure.

1.9, activating the Joule heaters causes the SVAs to shrink, and allowing the SVAs to cool causes them to expand. SVAs with different material properties, and therefore different deformation characteristics, can be produced, which enables the fabrication of heterogeneous hydrogel structures that can manipulate objects, avoid obstacles, generate traveling waves, and morph into different shapes. This research paves the way toward tunable, untethered, high-degree-of-freedom hydrogel-based robots that can adapt and respond to changing conditions in unstructured environments. The development of local actuators that can be distributed throughout a soft robot enables the implementation of decentralized control approaches on such robots, which would display robust performance to individual actuator failures.

We have experimentally validated our Cosserat rod dynamical model on a millimeter-scale soft robot arm composed of hydrogel SVAs (Doroudchi *et al.*, 2020). We have also experimentally identified and validated a linear state-space model of a miniature 2-DOF manipulator controlled by two hydrogel SVAs and applied an H_∞ output feedback approach for trajectory tracking (Doroudchi *et al.*, 2021).

1.3 Research Contributions and Organization of the Dissertation

The contributions of this dissertation can be summarized as follows. The publications cited below are listed at the end of this section.

We developed a novel practical approach to designing decentralized state feedback controllers for soft continuum robot arms composed of segments with local sensing, actuation, and control capabilities [1]. We used a semi-discrete numerical approximation of the Euler-Bernoulli beam equation, formulated in state space representation, to represent the robot arm dynamics. We designed three algorithms to solve an H_∞ optimal control problem, defined in terms of a Bilinear Matrix Inequality (BMI), by computing the tridiagonal feedback matrix which minimizes the H_∞ norm of the map from disturbances to regulated outputs. We confirmed through simulations that the controllers successfully dampen the free vibrations of a cantilever beam. This work is described in Section 2.1.

We also demonstrated, for the first time, the use of active temperature-responsive hydrogel actuators (Soft Voxel Actuators, or SVAs) as independently-controllable units for output-feedback control of a millimeter-scale 2-DOF manipulator in real-time for trajectory tracking [4]. We developed and experimentally identified a linear state-space model of the manipulator from input-output measurement data, which can be used to implement a variety of control techniques. This linear model is sufficiently accurate for control purposes, despite the complex nonlinear dynamics of the actuators. The Normalized Mean Absolute Error (NMAE) between the modeled and measured displacement of the manipulator's tip was below 10%. We demonstrated an exemplar payload transport application using an array of four manipulators with this versatile and computationally inexpensive technique. This work is described in Section 2.2.

To tackle the challenge of developing a dynamic model for continuum robots constructed from soft materials, we devised a novel decentralized control approach for configuration tracking by soft continuum robot arms composed of independently-controllable segments with local sensing and actuation [3]. We proposed a control approach that can be implemented on continuum robots with dynamics that can be described by the geometrically exact Cosserat rod model and enables such robots to track 3D configurations that involve any combination of bending, torsion, shear, and extension deformations. We validated this approach in numerical simulations of both hydrogel-based and silicone continuum robot arms, which tracked desired configurations with average normalized root-mean-square errors (NRMSEs) below 7%, indicating effective tracking performance. This work is described in Section 3.1.

Then, we experimentally identified and validated a Cosserat rod model of a millimeter-scale continuum robotic arm, composed of two rows of SVAs [2]. We compared simulations of the model to measurements of the robot's tip displacement over time during open-loop control trials in which subsets of the SVAs were actuated, producing bending deformations of the robot about a single axis. The NRMSEs between the simulated and experimentally measured displacements were below 10%. A novel experimental setup was used to identify the relationship between step input voltages applied to SVAs and their resulting force outputs during contraction and expansion, which resulted in a model of force over time with NRMSE less than 5%. This work is described in Section 3.2.

Lastly, we implemented our decentralized inverse dynamic control approach, based on a Cosserat rod model of a soft robotic arm, on an actual silicone multi-segment robotic arm for the first time [5]. The inverse dynamics problem was solved using a simulation of the Cosserat rod model, and the computed control output was applied to the actual robot in real-time. The complexity and computational cost of the controller are reduced by estimating unmeasured parameters using the simulated model. This approach was used to control the robot's configuration in 3D space for desired extension and bending deformations, which are useful for applications that require complex manipulation and locomotion. This work is described in Section 3.3.

[1] **Doroudchi, Azadeh**, Shivakumar, Sachin, E. Fisher, Rebecca, Marvi, Hamid, Aukes, Daniel, He, Ximin, Berman, Spring, and Peet, Matthew M.. “Decentralized Control of Distributed Actuation in a Segmented Soft Robot Arm." *2018 IEEE Conference on Decision and Control (CDC)*, pp. 7002-7009.(Oral presentation)

[2] **Doroudchi, Azadeh**, Khodambashi, Roozbeh, Lafmejani, Amir Salimi, Aukes, Daniel M., and Berman, Spring. “Dynamic Modeling of a Hydrogel-based Continuum Robotic Arm with Experimental Validation." *2020 IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 695-701. (Oral presentation)

[3] **Doroudchi, Azadeh** and Berman, Spring. “Configuration Tracking for Soft Continuum Robotic Arms Using Inverse Dynamic Control of a Cosserat Rod Model.” *2021 IEEE International Conference on Soft Robotics (RoboSoft).* (Poster presentation)

[4] **Doroudchi, Azadeh**, Khodambashi, Roozbeh, Sharifzadeh, Mohammad, Li, Dongting, Berman, Spring, and Aukes, Daniel. “Tracking Control of a Miniature 2-DOF Manipulator with Hydrogel Actuators.” *IEEE Robotics and Automation Letters (RA-L), March 2021 and the 2021 IEEE International Conference on Soft Robotics (RoboSoft).* (Oral presentation)

[5] **Doroudchi, Azadeh**, Qiao, Zhi, Zhang, Wenlong, and Berman, Spring. “Implementation of a Decentralized Tracking Controller on a Silicone-Based Soft Robotic Arm using Cosserat Rod Dynamics.” In preparation for *IEEE Robotics and Automation Letters (RA-L)*, 2022.

Chapter 2

CLOSED-LOOP CONTROL OF HYDROGEL-ACTUATED SOFT ROBOTS AND MECHANISMS BASED ON LINEAR STATE-SPACE MODELS

This chapter describes research published in Doroudchi *et al.* (2018, 2021).

2.1 Vibration Control of Soft Continuum Robot Arms Described by the Euler-Bernoulli Beam Model

In this section, we develop a novel practical approach to designing decentralized state feedback controllers for soft continuum robot arms composed of segments with local sensing, actuation, and control capabilities. The control objective is to regulate the robot arm's displacement in the presence of disturbance inputs; i.e., to dampen its disturbance-induced vibrations. Our approach does not require the use of a complex nonlinear model that describes the infinite-dimensional dynamics of the robot. Instead, we represent the robot arm's spatiotemporal dynamics using a semi-discrete numerical approximation of the Euler-Bernoulli beam PDE. This numerical approximation is formulated as an ordinary differential equation (ODE) state space model for implementation in linear matrix inequality (LMI) methods. The state space model is used to define an H_∞ optimal control problem in terms of a BMI. We present three algorithms of increasing stability and performance that solve this problem by computing the tridiagonal feedback matrix which minimizes the H_∞ norm of the map from disturbances to regulated outputs. Finally, we simulate the controllers computed by each algorithm for the case of a cantilever beam composed of hydrogel material and compare their performance.

2.1.1 Dynamic Model

The robot arm is constructed from N identical cylindrical segments that are arranged in a series configuration, as illustrated in Fig. 2.1. We assume that each segment is equipped with sensing, actuation, control, and communication elements. We also assume that each segment can apply local torques and can measure local deformations. For example, He *et al.* (2012) explain that when a segment is composed of force-sensitive conductive hydrogel, local deformations can be sensed from a resulting change in resistivity across the segment, and this change in resistivity provides an electrical signal which can be used as an output to a local feedback controller. The local controller can then induce a current, which causes local temperature changes in the segment that produce prescribed deformations and resulting torques.

In this decentralized sensing and actuation model of a soft robot arm, we likewise impose a decentralized communication architecture with a similar chain topology, meaning that each segment can exchange state measurements only with adjacent segments.

Model definition

To model the segmented robot arm, we will use a discretized version of the cantilever beam, wherein the beam is composed of material that is elastic, homogeneous, and isotropic. The beam is composed of material with Young's modulus E and density ρ. The beam has length L and a uniform cross-section of area A_c, and area moment of inertia I about the neutral axis.

Let $w(x,t)$ be the transverse displacement (see Fig. 2.1) of point $x \in [0,L]$ on the beam at time $t \in [0,T]$, where T is a specified final time. The PDE describing a one-dimensional unforced Euler-Bernoulli beam is given by

$$b^2 \frac{\partial^4 w}{\partial x^4} + \frac{\partial^2 w}{\partial t^2} = 0, \quad b^2 = \frac{EI}{\rho A_c}. \tag{2.1}$$

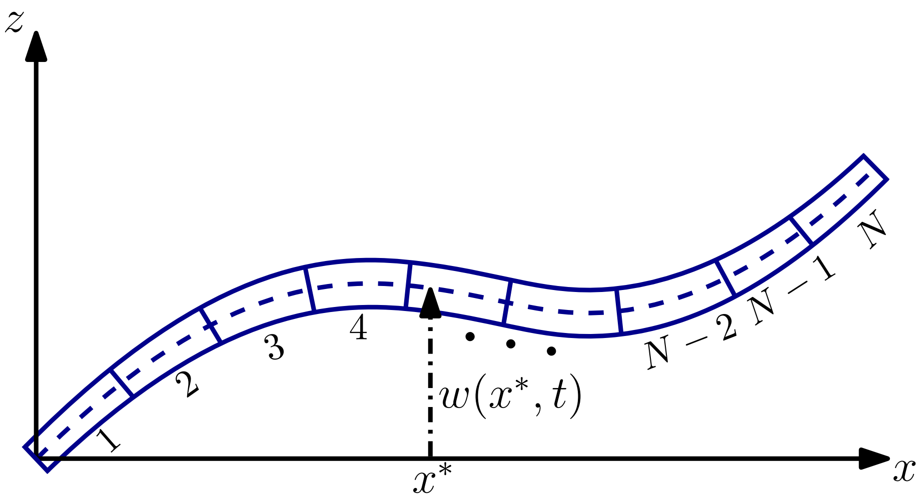


Figure 2.1: Semi-discrete beam model with N segments.

We define boundary conditions for this model that describe a cantilever beam, in which the deflection and slope of the fixed end and the bending moment and shear force at the free end are all set to zero:

$$\begin{aligned} w(0,t) &= 0, & \frac{\partial^2 w}{\partial x^2}(L,t) &= 0, \\ \frac{\partial w}{\partial x}(0,t) &= 0, & \frac{\partial^3 w}{\partial x^3}(L,t) &= 0, \end{aligned} \tag{2.2}$$

where $t \in [0, T]$.

We note that the Euler-Bernoulli beam equation is linear, assumes small shear stresses and is only accurate for small deflections. However, it has been shown by Bayo (1986) that for a uniform circular cross-section with diameter D, when $L \geq 20D$ the Euler-Bernoulli beam model yields a reasonably accurate approximation of the robot arm dynamics when the material properties satisfy certain assumptions. Furthermore, we note that the use of the robust control framework in this section mitigates the effect of inaccuracy in the model.

State space representation

To represent the segmented arm, we construct a discretized approximation of the continuum PDE beam model (2.1), (2.2), which results in a set of linear ODEs. As done by Bugariu *et al.* (2016), we also apply the central finite difference method with second-order accuracy to obtain a semi-discrete space approximation of model (2.1), (2.2) (discrete in the spatial coordinate x and continuous in time t). We define $h = L/N$ as the length of each segment and x_j as the x position of the right boundary of segment $j \in \{1, ..., N\}$. Then we have that $x_j = jh$ for each segment j, and we define $x_0 = 0$. For the boundary conditions, we also introduce two external points $x_{-1} = -h$ and $x_{N+1} = L + h$. The semi-discretization version of model (2.1) is then given by the following system of N linear equations, each describing the dynamics of the transverse displacement of point x_j on the beam at time $t \in [0, T]$:

$$\begin{aligned} \ddot{w}(x_j, t) = &-\frac{b^2}{h^4}[w(x_{j+2}, t) - 4w(x_{j+1}, t) + 6w(x_j, t) \\ &- 4w(x_{j-1}, t) + w(x_{j-2}, t)], \;\; j = 1, ..., N. \end{aligned} \tag{2.3}$$

Note that the dynamics of each segment's displacement is approximated as a function of its own displacement and that of the two closest segments on either side. The boundary conditions (2.2) are expressed as

$$\begin{aligned} &w(x_0, t) = 0, \;\; w(x_{-1}, t) = -w(x_1, t), \\ &w(x_{N+1}, t) = w(x_N, t), \;\; w(x_{N+2}, t) = w(x_{N-1}, t). \end{aligned} \tag{2.4}$$

We define the system state variables as $w(x_j, t)$, $\dot{w}(x_j, t)$, $j = 1, ..., N$ and arrange them in the vectors $w = [w(x_1, t) \; w(x_2, t) \; ... \; w(x_N, t)]^T$, $\dot{w} = [\dot{w}(x_1, t) \; \dot{w}(x_2, t) \; ... \; \dot{w}(x_N, t)]^T$. The system of equations (2.3) and the boundary conditions (2.4) can then be represented in state space form as follows:

$$\begin{bmatrix} \dot{w} \\ \ddot{w} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} w \\ \dot{w} \end{bmatrix} \tag{2.5}$$

where

$$\begin{aligned} A_{11} &= [0]_{N \times N}, \quad\;\; A_{12} = I_{N \times N}, \\ A_{21} &= \frac{-b^2}{h^4} A_h, \;\; A_{22} = [0]_{N \times N}, \end{aligned}$$

and the matrix $A_h \in \mathbb{R}^{N \times N}$ is defined as

$$A_h = \begin{bmatrix} 5 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ -4 & 6 & -4 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -4 & 6 & -4 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 1 & -4 & 6 & -4 & 1 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 1 & -4 & 6 & -3 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 1 & -3 & 2 \end{bmatrix}.$$

Including inputs and outputs, we obtain a state space representation given by

$$\begin{bmatrix} \dot{w} \\ \ddot{w} \end{bmatrix} = A \begin{bmatrix} w \\ \dot{w} \end{bmatrix} + Bu, \qquad y = C \begin{bmatrix} w \\ \dot{w} \end{bmatrix} + Du, \tag{2.6}$$

in which the system control input is denoted by $u \in \mathbb{R}^N$ and the output by $y \in \mathbb{R}^{2N}$. The A, B, C, and D matrices are defined as

$$\begin{gathered} A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}_{2N \times 2N}, \quad B = \begin{bmatrix} \begin{bmatrix} 0 \end{bmatrix}_{N \times N} \\ I_{N \times N} \end{bmatrix}_{2N \times N}, \\ C = I_{2N \times 2N}, \quad D = \begin{bmatrix} \begin{bmatrix} 0 \end{bmatrix}_{N \times N} \\ I_{N \times N} \end{bmatrix}_{2N \times N}. \end{gathered} \tag{2.7}$$

In Section 2.1.2, we discuss how the decentralized communication constraint leads to structural constraints on the gain from input to output.

2.1.2 Controller Synthesis

In this section, we use the linear ODE model developed in Section 2.1.1 to define a decentralized control problem assuming local full-state feedback. We first impose the mild assumption that the uncontrolled system is neutrally stable and controllable. To define the H_∞-optimal control problem, we use the standard regulator framework, yielding the 2-input, 2-output system representation $R \in \mathbb{R}^{7N \times 7N}$ as:

$$R = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right], \tag{2.8}$$

where

$$\begin{aligned} B_1 &= \begin{bmatrix} B & 0 \end{bmatrix}_{2N \times 4N}, \ B_2 = B, \\ C_1 &= \begin{bmatrix} C \\ 0 \end{bmatrix}_{3N \times 2N}, \quad C_2 = C, \\ D_{11} &= \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}_{3N \times 4N}, \ D_{12} = \begin{bmatrix} D \\ I \end{bmatrix}_{3N \times N}, \\ D_{21} &= \begin{bmatrix} D & I \end{bmatrix}_{2N \times 4N}, \ D_{22} = D. \end{aligned}$$

Because $C_2 = I$, the control problem is one of full-state feedback. The control problem, then, is to find the feedback controller $u = Ky$, $K \in \mathbb{R}^{N \times 2N}$, that minimizes the H_∞ norm of the map from disturbing inputs u to regulated outputs y. However, we now add a communication constraint in which we specify the structure of K to be tridiagonal (NOT block-diagonal). This structure implies that the moment generated by each segment is based only on measurements of its own state and the states of its two neighboring segments. Define the set of tridiagonal matrices as

$$T := \{K \in \mathbb{R}^{n\times n} \ : \ K = \begin{bmatrix} k_{1,1} & k_{1,2} & 0 & \cdots & 0 & 0 & 0 \\ k_{2,1} & k_{2,2} & k_{2,3} & \cdots & 0 & 0 & 0 \\ 0 & k_{3,2} & k_{3,3} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & k_{n-2,n-2} & k_{n-2,n-1} & 0 \\ 0 & 0 & 0 & \cdots & k_{n-1,n-2} & k_{n-1,n-1} & k_{n-1,n} \\ 0 & 0 & 0 & \cdots & 0 & k_{n,n-1} & k_{n,n} \end{bmatrix}, \ k_{i,j} \in \mathbb{R}\Big\} \tag{2.9}$$

We now denote the set of admissible controller gains by S, where

$$S := \{\, [K_1 \ K_2] \ | \ K_1, K_2 \in T \,\}.$$

This allows us to represent the controller information constraint as $K \in S$.

We may now formulate the H_∞ optimal control problem as a Bilinear Matrix Inequality (BMI). By using the bounded-real lemma from Boyd *et al.* (1994), it can be shown that γ is an H_∞ norm bound of the transfer function from input to output if there exists a positive definite matrix P and controller K which satisfy the BMI constraint (2.10). Consequently, we can formulate the optimization problem (2.10) below, whose solution is the H_∞-optimal decentralized controller $K^* \in S$.

minimize $\gamma > 0$ such that

$$\begin{bmatrix} (A + B_2K)^T P + P(A + B_2K) & *^T & *^T \\ B_1^T P & -\gamma I & *^T \\ (C_1 + D_{12}K) & D_{11} & -\gamma I \end{bmatrix} < 0 \tag{2.10}$$

for some $K \in S$ and $P > 0$.

In the matrix inequality, "$*$" is used to represent symmetric elements of the matrix.

2.1.3 Proposed Algorithms for Solving the BMI

The optimization problem (2.10) is a BMI in the matrix variables K and P. Solving BMIs is known to be an NP-hard problem as declared by Toker and Ozbay (1995). In this section, we evaluate three possible algorithms for obtaining locally optimal solutions to this BMI, two based on iteration and one based on gradient descent.

2.1.4 Initialization

In all three algorithms, we require an initial feasible solution to the BMI. Furthermore, the selection of initial values can significantly influence convergence to an optimal solution. Unfortunately, however, there are no canonical rules for finding an initial feasible solution.

In our algorithm, we address this problem as follows. Under the assumption that the nominal system is controllable, the following LMI has solution $P > 0$:

$$\text{controllability:} \quad A^T P + PA - BB^T < 0 \tag{2.11}$$

We use this solution as an estimate of the initial value of P (P_0). Using this P_0 to find the initial value of K (K_0) is problematic, however, because of the additional constraint $K \in S$. To resolve this, we initialize K without the sparsity constraint and solve the resulting LMI version of (2.10) for P, and then use this as our new estimate of P_0. Given this new value of P_0, we solve the resulting LMI version of (2.10) for K with the relaxed constraint that only the last row of K is required to have the sparsity structure $K \in S$ of the tridiagonal matrix (2.9). Using this K, we solve the LMI version of (2.10) for P. This procedure is repeated by progressively constraining more rows of K to have the structure of the corresponding row of matrix (2.9) until the entire matrix K has the desired tridiagonal structure.

We have developed the following three algorithms to obtain an H_∞ optimal solution for

K. The algorithms all use the initialization procedure described above.

Iterative optimization algorithm

Algorithm 1 is a standard iteration-based method used to solve a bilinear system of equations. It is similar to our initialization procedure for the variables P and K. Initializing a value for P (P_0) yields the LMI from (2.10), which is solved by optimizing over K. Afterward, we fix K in (2.10) and optimize over P. These steps are repeated until the values of K and P converge to optimal values, at which point the change in γ is minimal. This algorithm has two drawbacks: it does not converge for certain initial values of P (P_0), especially if the A matrix is numerically ill-conditioned, and the solution for K could have a large magnitude that makes the feedback controller physically impractical to implement. However, imposing additional constraints on the magnitude of K could potentially cause the H_∞ norm to diverge. We next propose two modified versions of this algorithm that address these problems.

Modified iterative optimization algorithm

Algorithm 1, depending on the choice of P_0, can end up oscillating between suboptimal solutions for K. This was observed to happen for poor choices of P_0. To reduce these oscillations, we define P and K at each iteration as weighted averages of their current values and their optimized values, obtained by solving the optimization problem (2.10) during the current iteration. The weight factor α is chosen to be a value between 0 and 1. The α value can be selected to produce small changes in the solution between iterations, thus preventing the solution from making large jumps in the non-convex subspace. An α close to 0 would result in very small changes in P and K over successive iterations.

Gradient descent algorithm

Although both Algorithms 1 and 2 are quick to converge, they do not converge at all when the matrices A and B are numerically ill-conditioned. In addition, the solution for K computed by these procedures often has a magnitude that is too large for implementation in practice. We address this problem in Algorithm 3 by splitting optimization problem (2.10) into two optimization problems with LMI constraints, shown below. The difference here is that we can directly restrict changes in the solution over successive iterations and also limit the values taken by the variables. We redefine the optimization variables as $\Delta K \in \mathbb{R}^{N\times 2N}$ and $\Delta P \in \mathbb{R}^{2N\times 2N}$, whose L_∞-norms are constrained to be small in order to prevent large changes in K and P between iterations.

minimize $\gamma_a > 0$ such that $\|\Delta K\| < \epsilon_1$ and

$$\begin{bmatrix} (A+B_2K_a)^TP + P(A+B_2K_a) & *^T & *^T \\ B_1^TP & -\gamma_a I & *^T \\ (C_1 + D_{12}K_a) & D_{11} & -\gamma_a I \end{bmatrix} < 0 \qquad (2.12)$$

for some $K_a \in S$, where $K_a \equiv K + \Delta K$.

minimize $\gamma_b > 0$ such that $\|\Delta P\| < \epsilon_2$ and

$$\begin{bmatrix} (A+B_2K)^TP_a + P_a(A+B_2K) & *^T & *^T \\ B_1^TP_a & -\gamma_b I & *^T \\ (C_1 + D_{12}K) & D_{11} & -\gamma_b I \end{bmatrix} < 0 \qquad (2.13)$$

for some $P_a > 0$, where $P_a \equiv P + \Delta P$.

In these two problems, ϵ_1 and ϵ_2 are small positive numbers.

The optimization procedure is performed alternately over ΔK and ΔP as follows such that γ converges to a local minimum. At the beginning of each iteration, problem (2.12)

Algorithm 1 Standard iterative algorithm

```
1: Choose a small $\epsilon > 0$. Initialize $P$ to $P_0$.
2: while $|\gamma_k - \gamma_{k-1}| > \epsilon$ do
3:     Use the last known value for $P$.
4:     Solve for $K$ in problem (2.10), minimizing $\gamma$.
5:     Use the solution for $K$ in the next step.
6:     Solve for $P$ in problem (2.10), minimizing $\gamma$.
7:     $\gamma_k$ is the minimized value of $\gamma$ in step 6.
8:     $k = k + 1$
9: end while
```

is solved for ΔK using the current values of K and P, and then K is increased by ΔK. Next, problem (2.13) is solved for ΔP, and P is increased by ΔP.

2.1.5 Simulation Results

We validated our numerical approximation of the beam model and investigated the performance of our decentralized state feedback controllers in simulation. *YALMIP* from Lofberg (2004), an optimization toolbox for MATLAB with the *MOSEK* solver from ApS (2017), was used to solve the optimization problems in Algorithms 1, 2, and 3. The beam model was simulated using the MATLAB *lsim* command with $N = 40$ segments and the parameters listed in Table 2.1, where E and ρ are defined for hydrogel material[1].

Validation of semi-discrete approximation of beam model

In order to evaluate the accuracy of the numerical approximation (2.3), (2.4), we compare it to the analytical solution of the cantilever Euler-Bernoulli beam model (2.1), (2.2). The

[1] N-isopropylacrylamide, variously abbreviated PNIPA, PNIPAAm, NIPA, PNIPAA or PNIPAm

Algorithm 2 Modified iterative algorithm

```
1: Choose a small $\epsilon > 0$ and $\alpha \in (0, 1)$. Initialize $P$ to $P_0$.
2: while $|\gamma_k - \gamma_{k-1}| > \epsilon$ do
3:     Use the last known value for $P$.
4:     Solve for $K$ in problem (2.10), minimizing $\gamma$.
5:     $K_{k+1} = K_k + \alpha(K - K_k)$
6:     Use $K_{k+1}$ as the current value of $K$.
7:     Solve for $P$ in problem (2.10), minimizing $\gamma$.
8:     $P_{k+1} = P_k + \alpha(P - P_k)$
9:     $\gamma_k$ is the optimal value of $\gamma$ in step 7.
10:    $k = k + 1$
11: end while
```

initial conditions of the beam model were set to

$$w(x, 0) = \sin\left(\frac{3\pi}{2L}x\right), \quad \frac{\partial w}{\partial t}(x, 0) = 0, \quad x \in [0, L]. \tag{2.14}$$

For these initial conditions, the solution to the beam model (2.1), (2.2) can be obtained using the separation of variables method:

$$w(x, t) = \sin\left(\frac{3\pi}{2L}x\right)\cos\left(\frac{9\pi^2 b}{4L^2}t\right), \quad t \in [0, T], \ x \in [0, L]. \tag{2.15}$$

This solution describes the first mode shape of the beam. In the simulations, we set $T = 20$.

Figures 2.2(a),(b) plot the vibrations of the beam over time from the analytical solution (2.15) and the numerical approximation, respectively. Figure 2.2(c) plots the error between the analytical solution and the numerical approximation. Although this error grows over time due to numerical approximation error propagation, it remains relatively small (the magnitude does not exceed 0.08 m within the first 5 s) compared to the maximum amplitude of the beam vibrations within the first few seconds of the simulation, when the controllers

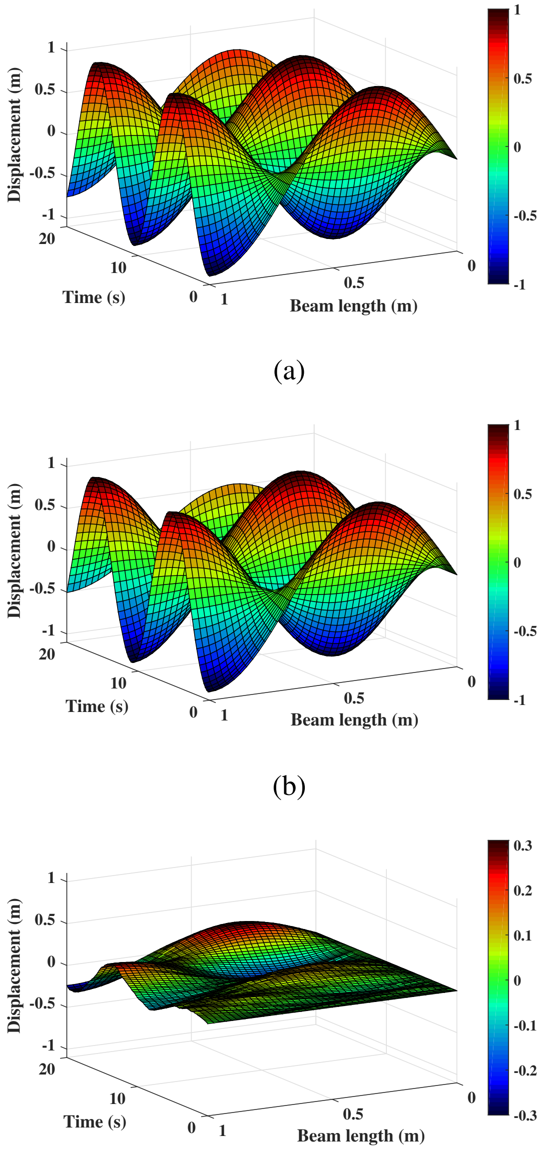
Algorithm 3 Gradient descent algorithm

```
Choose a small $\epsilon > 0$. Initialize $K$ to $K_0$ and $P$ to $P_0$.
while $|\gamma_k - \gamma_{k-1}| > \epsilon$ do
    Solve problem (2.12), minimizing $\gamma_a$.
    $K = K + \Delta K$
    Solve problem (2.13), minimizing $\gamma_b$.
    $P = P + \Delta P$
    $\gamma_k$ is the optimal value of $\gamma_b$.
    $k = k + 1$
end while
```

Table 2.1: Beam Material and Geometric Properties

Parameter	Definition	Value	Units
E	Young's modulus at $25°C$	5.0	kPa
ρ	Mass density	1.1	g/cm^3
D	Diameter	5.0	cm
L	Length	1.0	m
A_c	Cross-section area	19.6	cm^2
I	Area moment of inertia	30.7	cm^4

effectively damp the vibrations (see next section). Thus, the numerical approximation is sufficiently accurate for use in our optimization methods to synthesize the controllers. Note that a relatively coarse spatial discretization ($N = 40$ segments) was used for the numerical approximation; a closer match to the analytical solution could be achieved with a finer discretization.



(a)

(b)

(c)

Figure 2.2: Euler-Bernoulli cantilever beam free vibrations from (a) the analytical solution (2.15) and (b) the numerical approximation (2.3), (2.4); (c) error between the analytical solution and numerical approximation.

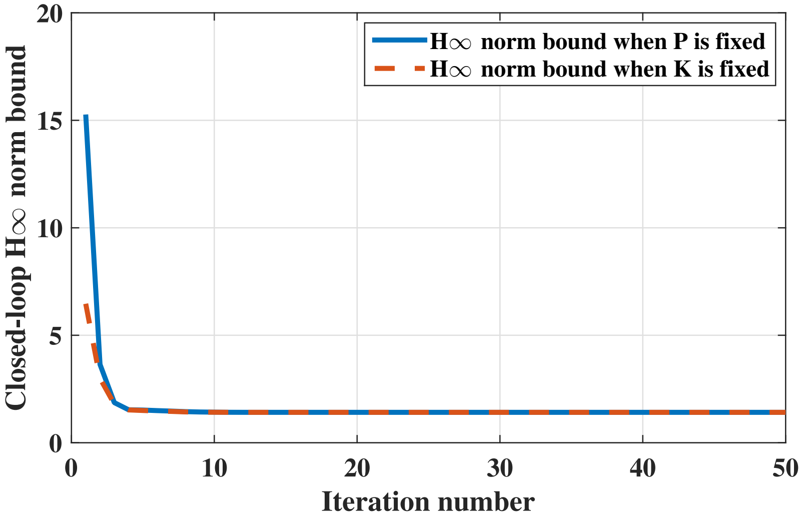
Comparison of optimal decentralized controllers for damping beam vibrations

Decentralized state feedback controllers were synthesized with Algorithms 1, 2, and 3, and the beam dynamics were simulated for each controller using the numerical approximation (2.6). All the variables were initialized using the procedure described in Section 2.1.4.

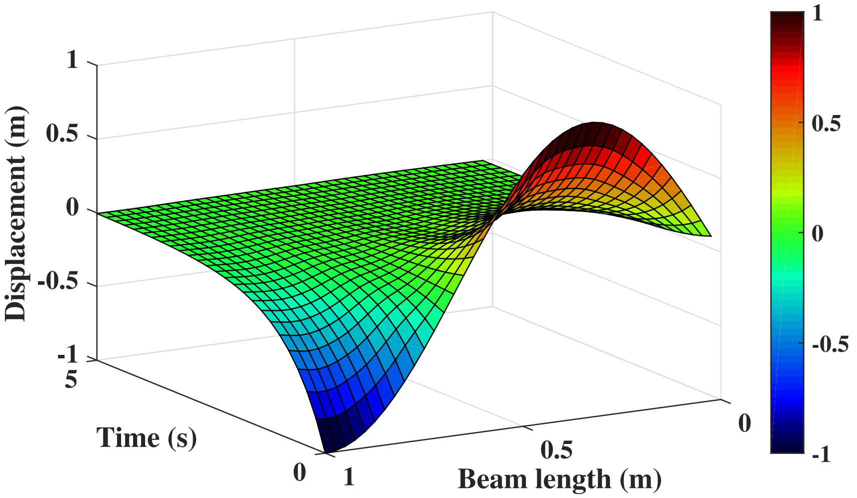
Figures 2.3a, 2.4a, and 2.5a plot the evolution of the closed-loop H_∞ norm bound, γ, over the execution of each algorithm when the optimization is performed alternately over the variables K and P during each iteration. Figures 2.3b, 2.4b, and 2.5b display the resulting closed-loop beam response for each controller given the initial condition (2.14). These figures show that all controllers successfully dampen the beam vibrations that are induced by the initial beam displacement within the first 5 seconds. From the convergence rates of the plots in Fig. 2.3a, 2.4a, and 2.5a, it is evident that Algorithm 1 is the least computationally intensive procedure, followed by Algorithm 2 and then by Algorithm 3. This is because K is least constrained in Algorithm 1, which therefore permits large changes in K between iterations and hence has the fastest convergence, followed by the other two algorithms. Algorithm 2 shows superior performance to Algorithm 1, in that it converged to a controller with a smaller H_∞ norm bound ($\gamma = 1.05$, versus $\gamma = 1.42$ for Algorithm 1) at the expense of a slight increase in computational demands. Algorithm 3 converged to the highest H_∞ norm bound ($\gamma = 2.31$) of the three methods since the controller gain values were subject to additional constraints. However, the controller computed by this algorithm would be the most feasible one to implement in practice, since the constraints limit the magnitudes of the controller gains.

2.1.6 Discussion

In this section, we developed three algorithms for synthesizing a decentralized controller for the discretized Euler-Bernoulli beam model by solving an H_∞ optimal control

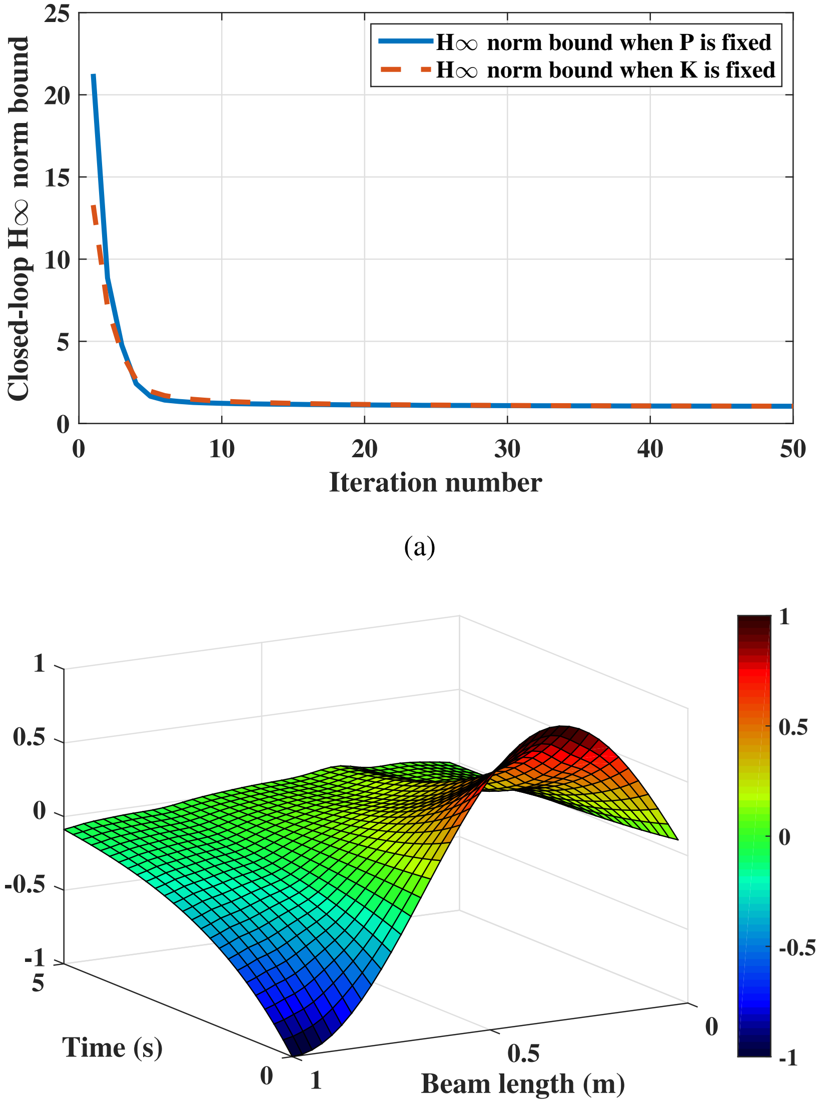


(a)



(b)

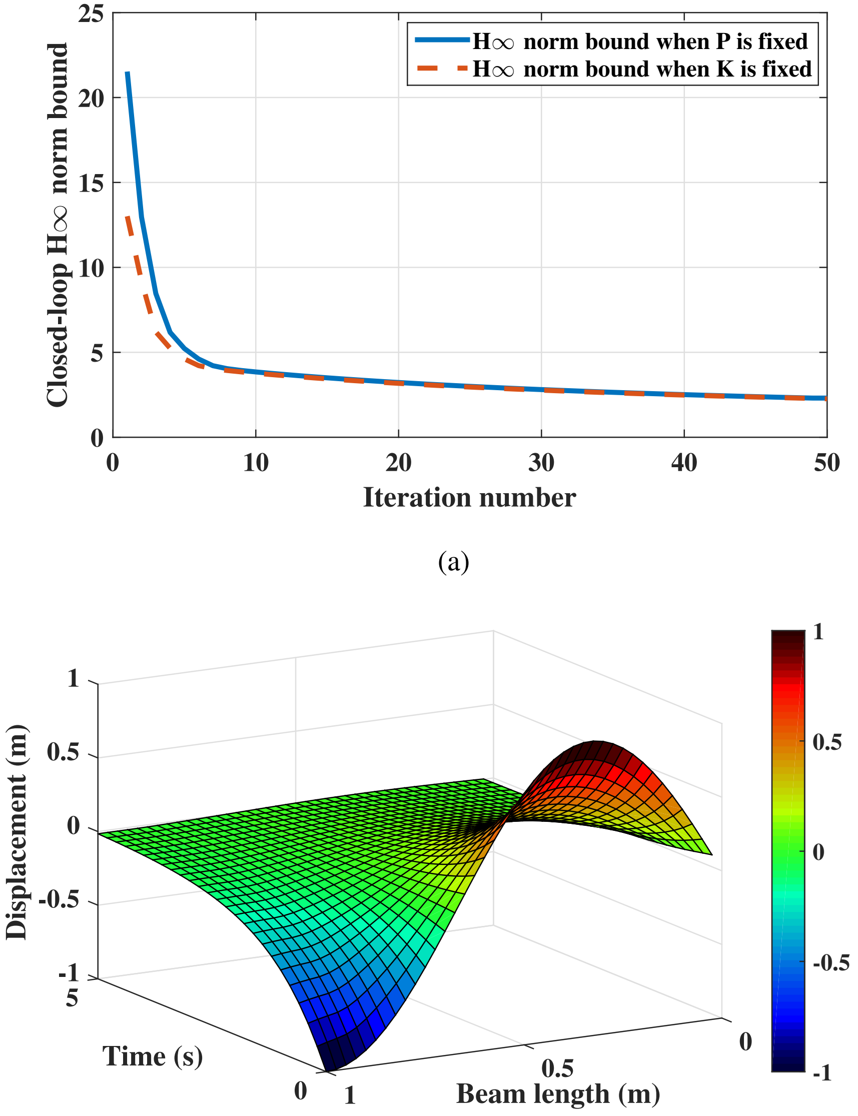
Figure 2.3: (a) H_∞ norm bound converging in the two alternating steps of Algorithm 1. (b) Closed-loop response of the simulated beam with initial conditions (2.14) and the controller from Algorithm 1.



(a)

(b)

Figure 2.4: (a) H_∞ norm bound converging in the two alternating steps of Algorithm 2. (b) Closed-loop response of the simulated beam with initial conditions (2.14) and the controller from Algorithm 2.



(a)

(b)

Figure 2.5: (a) H_∞ norm bound converging in the two alternating steps of Algorithm 3. (b) Closed-loop response of the simulated beam with initial conditions (2.14) and the controller from Algorithm 3.

problem. We found that when the system matrix is numerically ill-conditioned, which is a common property of discretized beam models, convergence of the H_∞ norm is not always guaranteed. In addition, we found that iterative approaches are in general sensitive to the initial selections of P and K. The modifications proposed in the algorithms solved these problems of convergence and sensitivity for the discretized beam model. The iterative and modified iterative methods quickly reach a converged H_∞ norm value, but they do not guarantee convergence for different selections of initial P and K. The gradient descent approach, while slightly slower at reaching a converged H_∞ norm value, is less sensitive to different choices of initial P and K. It provides a bounded solution for the controller gains, which is often a necessity in physical systems.

2.2 Development and Validation of a Tracking Controller for a Miniature 2-DOF Hydrogel-Actuated Manipulator

Due to the nature of the complex spatiotemporal dynamics of stimuli-responsive soft materials, closed-loop control of hydrogel-actuated mechanisms has remained a challenge. This section demonstrates, for the first time, closed-loop trajectory tracking control in real-time of a millimeter-scale, two degree-of-freedom manipulator via *independently-controllable*, temperature-responsive hydrogel actuators. Our ability to coordinate independently controllable soft actuators with complex internal dynamics in a robotic system demonstrates progress in the real-time, closed-loop control of mechanisms with this type of actuator. We expect that researchers will be able to adapt this approach across similar stimuli-responsive materials as they are developed and optimized. This will also permit SVAs, manufactured from a variety of materials, to be used for controlled grasping, manipulation, and locomotion tasks across a variety of new soft robotic platforms, such as our octopus-inspired continuum robot described in Section 3.2.

A linear state-space model of the manipulator is developed from input-output measurement data, enabling the straightforward application of control techniques to the system. The Normalized Mean Absolute Error (NMAE) between the modeled and measured displacement of the manipulator's tip is below 10%. We propose an Observer-based controller and a robust H_∞-optimal controller and evaluate their performance in a trajectory tracking output-feedback framework, compared with and without sinusoidal disturbances and noise. We demonstrate in simulation that the H_∞-optimal controller, which is computed using Linear Matrix Inequality (LMI) methods, tracks an elliptical trajectory more accurately than the Observer controller and is more robust to disturbances and noise. We also show experimentally that the H_∞-optimal controller can be used to track different trajectories with an NMAE below 15%, even when the manipulator is subject to a 3 g load, 12.5 times an actuator's weight. Finally, a payload transport scenario is presented as an exemplar application; we demonstrate that an array of four manipulators is capable of moving a payload horizontally by applying the proposed H_∞-optimal trajectory-tracking controller to each manipulator in a decoupled manner.

We demonstrate our control approach on a millimeter-scale, two degree-of-freedom (DOF) manipulator actuated by two SVAs, shown in Fig. 2.6. Many prior control-oriented models developed for similar systems have been governed by the kinematic equations describing rigid links reviewed by Webster III and Jones (2010c), which are less useful in the design of feedback controllers for continuously deformable robots with soft actuators embedded within their structure. To address this, we propose a black-box identified model based on methods in Ljung (2010) that simplify system dynamics in the form of a linear state-space representation. Lim *et al.* (1998) point out that modern control techniques, being built upon state-space models and state-space system identification, are more practical in application. We apply system identification methods to obtain a linear state-space model of the manipulator, which can be used to implement a wide range of controllers for different applications. We design an H_∞-optimal output-feedback tracking controller, similar to the

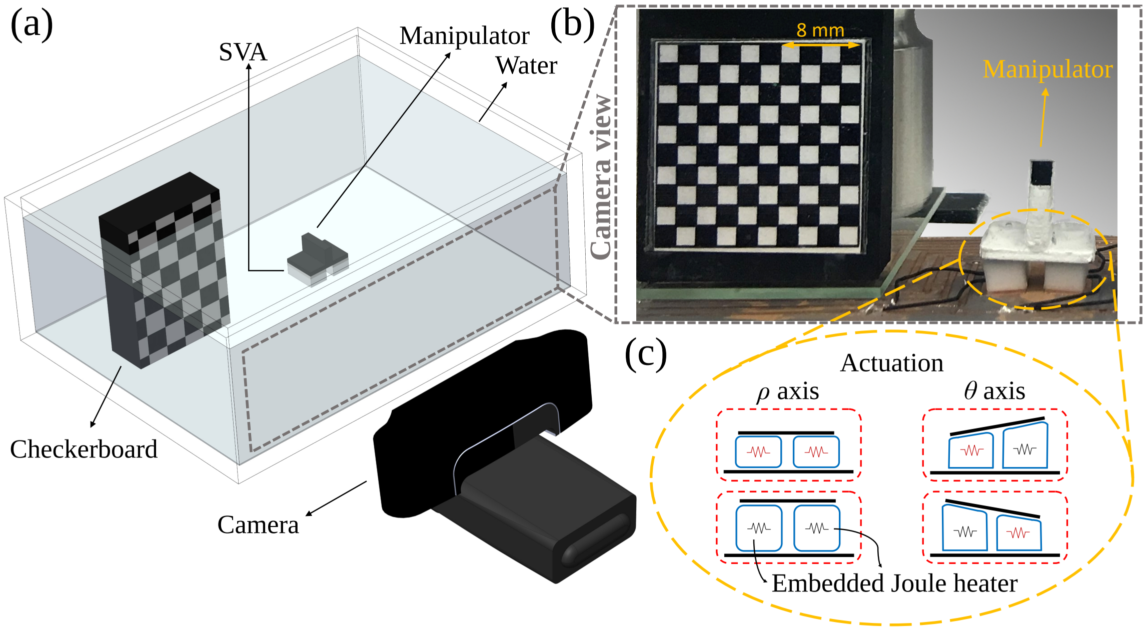


Figure 2.6: Experimental setup for tracking control of a 2-DOF manipulator with embedded hydrogel Soft Voxel Actuators (SVAs) (Section 3.2). (a) Illustration of a manipulator in a water-filled tank with a camera for vision-based feedback of the manipulator tip. (b) Camera view of the setup, including a fabricated manipulator prototype and checkerboard for camera calibration. (c) Illustration of SVA deformation in various activation states (red = on; black = off).

H_∞ output-feedback controller proposed by Farhamfard *et al.* (2016) for flexible needle guidance with the difference that their control system is dynamic rather than static. We then compare it in simulation to an observer-based output-feedback controller. The H_∞-optimal controller is then experimentally validated for planar reference trajectories. Finally, we show that our approach can be used to control more complex mechanisms actuated by SVAs through a demonstration of payload transport by four manipulators.

In summary, the contributions of this section are as follows:

- Implementation of active temperature-responsive hydrogel-based actuators (the SVA) as

independently-controllable units.

- Development and experimental identification of a linear state-space model of the manipulator that can be used to implement a variety of control techniques. This linear model is sufficiently accurate for control purposes, despite the complex nonlinear dynamics of the actuators.
- Demonstration of, for the first time, the ability to control a 2-DOF mechanism with **independently-controllable** hydrogel actuators in real time using output-feedback controllers.
- Demonstration of an exemplar payload transport application using an array of four manipulators with this versatile and computationally-inexpensive technique.

2.2.1 Manipulator Fabrication

SVAs are fabricated by embedding small Joule heaters within a mold, temperature-responsive PNIPAAM hydrogel in the shape of a rectangular prism, as illustrated in Fig. 2.6b and 2.6c. When an electric potential is applied across the embedded Joule heater, the actuator shrinks uniformly. The manipulator, also shown in Fig. 2.6b, consists of two SVAs affixed to a 3D-printed T-shaped extension, which serves as the end-effector. A standard PNIPAAM hydrogel precursor solution is used to fabricate the SVAs from thermo-responsive hydrogel, using a recipe described by Khodambashi *et al.* (2021). Each SVA is $8 \times 4.5 \times 3\,\mathrm{mm}^3$ in its fully swollen state, with a total weight of $0.12\,\mathrm{g}$, including the embedded-Joule heater ($10\,\Omega$ SMD resistor 0805), which is connected to microcontrollers by wires. The T-shaped extension is 3D-printed in nylon using a Markforged M2 3D printer. A circuit board, which serves as the fixed base of the manipulator, is attached to one side of the two SVAs; the T-shaped extension is attached to the other side. The circuit board and extension are attached to the SVAs with superglue to ensure that they remain in contact with the SVAs during the experiments. Since hydrogels must be immersed in water to absorb water when cooling, all experiments are conducted in a tank of deionized (DI), room-temperature water.

2.2.2 Experimental Setup

Figure 2.6 shows the experimental setup used for closed-loop control and tracking of the manipulator's trajectory. A Logitech C930e USB Webcam is placed in front of the tank to send real-time data to the image processing program in MATLAB which tracks the position of a marker on the manipulator tip. These measurements of the manipulator tip's position over time are transmitted back to the controller. We used a black-and-white checkerboard with 2 mm $\times$ 2 mm squares to estimate the camera calibration factors (mm/pixel) along the x and y axis (Fig. 2.6). White was selected as the color of the tank's background, and black was selected as the color of the manipulator tip's marker to facilitate contrast-based filtering between the foreground and background. The Camera Calibration Toolbox in MATLAB was initially used to compensate for lens distortion, but since this increased the image processing time by 30% without significantly improving the image data, the original camera images were subsequently used without compensation. All control algorithms are implemented in MATLAB; the controller output is sent to an Arduino Mega2560, which acts as the physical communication layer between MATLAB and a PCA9685 MOSFET board. This MOSFET board, with 16 discrete output channels, receives a PWM signal from the controller and applies it (maximum: 3.7 V) at higher current to the corresponding Joule heater.

2.2.3 Manipulator Modeling

In this section, the kinematics of the manipulator are derived in order to compute its workspace. A two-dimensional linear state-space model of the manipulator is then defined

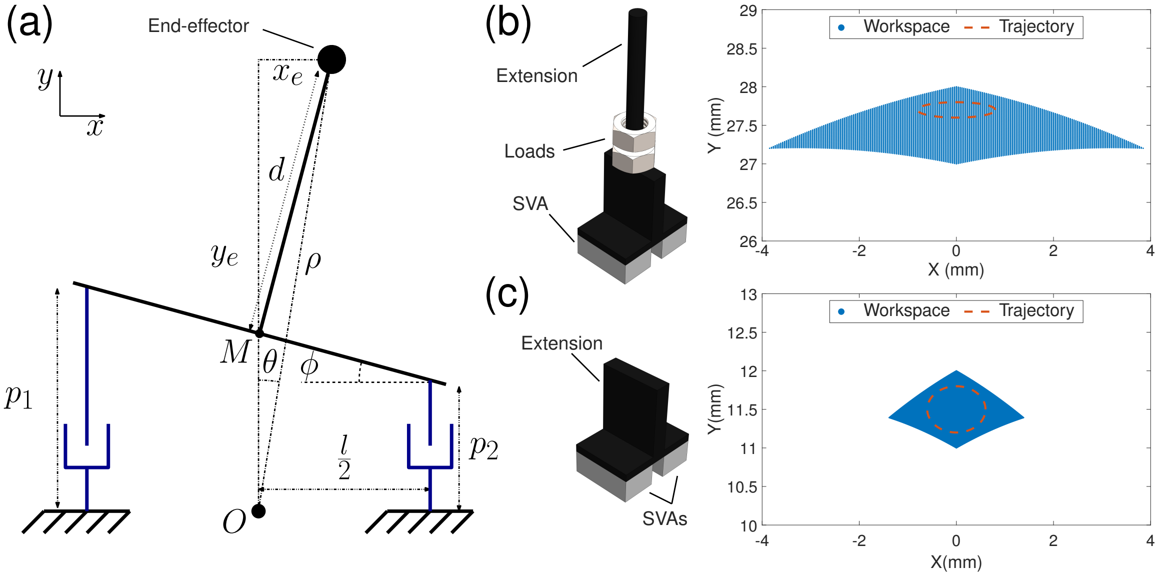


Figure 2.7: Schematic and workspace of the manipulator. (a) The 2-DOF parallel mechanism model with two prismatic joints. (b), (c) Workspaces for extensions with lengths $d = 25\,\text{mm}$ and $d = 9\,\text{mm}$, respectively, with exemplar elliptical trajectories overlaid in red. Extra loads are added to the longer extension in (b) to test the robustness of the controller in experiment.

using black-box system identification methods.

2.2.4 Kinematics and Workspace

To derive manipulator tip kinematics and workspace, the SVAs are modeled as two prismatic joints, as illustrated in Fig. 2.7a. The T-shaped extension is assumed to be rigid compared to the SVAs that are rigidly attached to the extension and circuit board. The SVAs are modeled as linear contractile elements since only one dimension of their volumetric shape change influences the displacement of the manipulator. Thus, with two prismatic actuators connected in parallel, the manipulator may be considered a 2-DOF mechanism. As shown in Fig. 2.7a, p_1 and p_2 are defined as the linear height of each SVA. These values vary between 3 mm in inactivated SVA to 2 mm when activated by the embedded Joule heaters; l =6.5 mm is the spacing between SVAs, d is the length of the extension, and ϕ shows the extension's angle from the horizontal axis. We assume point M's displacement in the x direction is negligible (Fig. 2.7a). The forward kinematics of the manipulator may be computed geometrically for the manipulator tip's, $\boldsymbol{p}_e$, in Cartesian coordinates, x_e and y_e, in the reference frame with origin O, according to

$$\boldsymbol{p_e} = \begin{bmatrix} x_e & y_e \end{bmatrix}^T, \ \ \phi = \arctan(\frac{p_1 - p_2}{l}), \tag{2.16}$$

$$x_e = d\sin(\phi), \ \ y_e = d\cos(\phi) + \frac{p_1 + p_2}{2}. \tag{2.17}$$

The polar coordinates ρ and θ of the manipulator's tip in this reference frame (see Fig. 2.7a) are given by

$$\rho = \sqrt{x_e^2 + y_e^2}, \ \ \theta = \arctan\left(\frac{x_e}{y_e}\right). \tag{2.18}$$

As illustrated in Fig. 2.6c, if both SVAs are activated simultaneously with the same input voltage, then the manipulator's tip moves along the ρ-axis at a constant θ; if only the left or right SVA is activated, then the tip undergoes an angular displacement at a constant ρ. Other SVA activation patterns produce a combination of displacements in both ρ and θ. Two different extensions with lengths of $d = 25$ mm and $d = 9$ mm were fabricated and tested, and their workspaces are shown in Figs. 2.7b and c, respectively. The longer extension is used to amplify the motion of each actuator, resulting in a larger workspace and making the controller performance easier to measure and evaluate. Extra loads may be added to the shaft of the longer extension, as shown in the left image in Fig. 2.7b, in order to experimentally test the robustness of the controller. The shorter extension, by contrast, supports higher loads on the tip during trajectory tracking, as demonstrated in the payload transport application in Section 2.2.12.

2.2.5 Linear State-Space Model

As explained in the last section, the displacements of the SVAs and manipulator tip are not decoupled, since the T-shaped extension connecting the actuators to the tip establishes a rigid kinematic transformation from the prismatic motion of the actuators to the 2-DOF planar motion of the end-effector. To find and select a model that best represents the system's behavior, a number of models were considered including state-space models of different dimensionalities. We model this control system using a two-dimensional linear state-space representation, which enables the implementation of a variety of control methods. Defining $\boldsymbol{x}(t) \in \mathbb{R}^{4\times1}$ as the vector of unknown system state variables at time t, $\dot{\boldsymbol{x}}(t)$ as the vector of time derivatives of the state variables, $\boldsymbol{u}(t) = [V_1(t) \quad V_2(t)]^T$ as the vector of inputs, and $\boldsymbol{y}(t) = [\theta(t) \quad \rho(t)]^T$ as the vector of outputs, the state-space model is given by

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) \end{aligned} \tag{2.19}$$

where the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ must be determined for each extension (25 mm and 9 mm), separately. Since the state variables of the model are not necessarily measurable, Lim *et al.* (1998) discuss that it is crucial to understand the relationship between various input-output models and state-space models in order to accurately identify the state-space model from input-output data. The 2-input 2-output state-space model showed a good fit to the data and also directly provides the unknown matrices that are required for designing the controller. $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are identified by applying black-box system identification to a set of time series input-output data according to method presented by Ljung (2001a), using the MATLAB System Identification Toolbox. The identified matrices for the 25 mm extension were found to be:

$$\boldsymbol{A} = \begin{bmatrix} -0.0007 & -0.0301 & 0.0444 & 6.0548 \\ -0.0016 & -0.0623 & 0.0254 & -1.4325 \\ -0.2613 & 0.6580 & 7.2633 & -374.9846 \\ -0.0243 & 0.1643 & 3.0590 & -44.3024 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 0.0001 & 0.0003 \\ -0.0000 & -0.0001 \\ -0.0051 & -0.0232 \\ -0.0001 & -0.0042 \end{bmatrix},$$

$$\boldsymbol{C} = \begin{bmatrix} 1.1446 & -0.0046 & -0.0020 & 0.0034 \\ -1.1431 & -3.5368 & 0.0020 & -0.0534 \end{bmatrix},$$

$$\boldsymbol{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Multiple input-output data sets were gathered across various ranges of amplitudes and frequencies to find the unknown matrices. Since the hydrogel-based SVAs have a relatively slow response compared to electric motors and other actuators, and it has a specific range of 25° to 32° from cooling to heating phases, the model is linearized around the fastest signal that permitted the hydrogel actuators to respond across their full temperature range. Figure 2.9a plots two selected input voltages among the data set that were experimentally applied to the SVAs which covers the tip workspace and depicts a 50% shift in the SVAs' input signal. The signal shift covers the actuation limits in the modeling and provides an active cooling phase which improves both the speed and tracking performance of the manipulator; it means that the SVAs were actuated from a starting point of half-actuation (50%) and then the signal reaches the minimum and maximum values during the cycle and accordingly the SVA reaches the maximum and minimum volume. Fig. 2.9b displays the

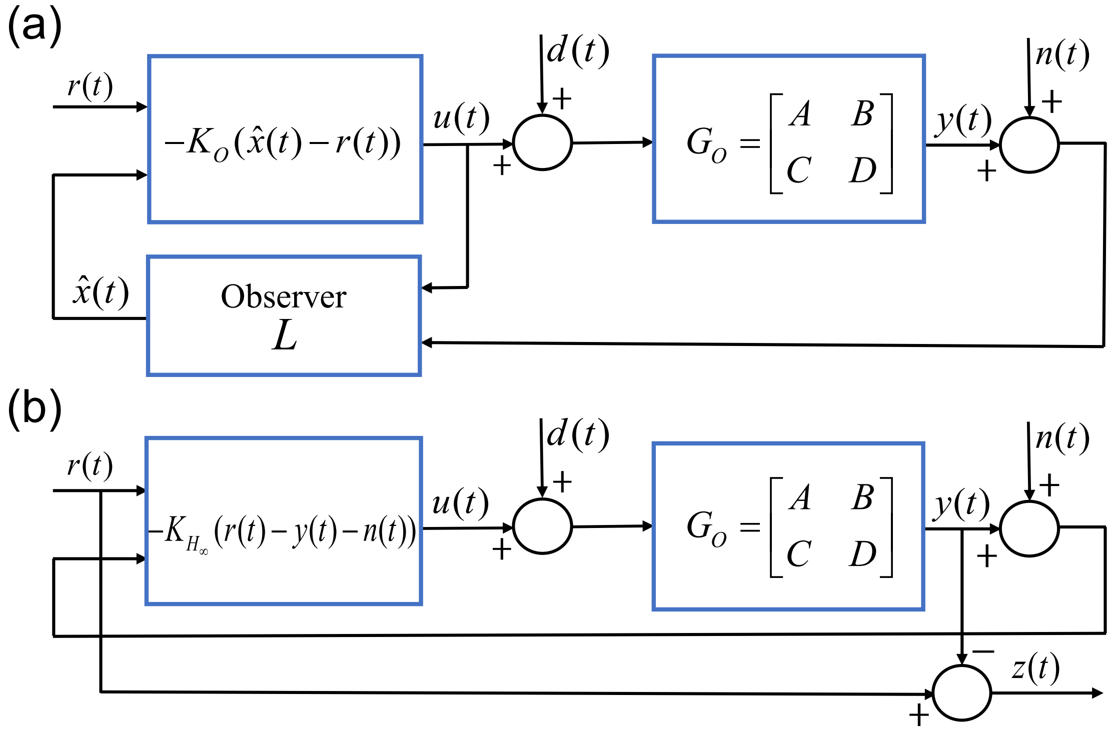


Figure 2.8: Block diagrams of the proposed output-feedback controllers with state-space representation in the tracking framework. (a) Observer-based controller. (b) H_∞-optimal controller.

resulting displacement of the manipulator tip and the outputs of the identified model (2.19) for the same inputs depicted in Fig. 2.9a. These figures are a comprehensive example of comparison between the actual data and the identified model output which show that the model outputs ρ and θ follow the corresponding measured output values throughout the duration of the experiment with sufficient accuracy. The average NMAE for ρ and θ in three repeating cycles from the plotted inputs in Fig. 2.9a are given by 5.5% and 7.5%, respectively. The NMAE value remains below 10% for other tested data sets. Thus, our linear state-space model of the entire mechanism is sufficiently accurate to use in the design of controllers for the manipulator, despite the difficult-to-characterize nonlinear dynamics of the hydrogel actuators themselves.

2.2.6 Controller Design

It can be shown that the open-loop state-space model (2.19), which has the corresponding transfer function $\boldsymbol{G}_o(s)$, is stable, controllable, and observable. In this section, we design two trajectory tracking controllers based on this state-space model, an observer-based output-feedback controller and an H_∞-optimal output-feedback controller. Block diagrams of the controllers are illustrated in Fig. 2.8. Both controllers are designed to track a reference trajectory $\boldsymbol{r}(t) \in \mathbb{R}^{2\times 1}$ while attenuating the effects of noise, denoted by $\boldsymbol{n}(t) \in \mathbb{R}^{2\times 1}$, and external disturbances, denoted by $\boldsymbol{d}(t) \in \mathbb{R}^{2\times 1}$.

2.2.7 Observer-Based Output-Feedback Controller

In this type of controller, an observer is designed to compute an estimate $\hat{\boldsymbol{x}}(t)$ of the unknown system state vector $\boldsymbol{x}(t)$ from the control input $\boldsymbol{u}(t)$ and the output $\boldsymbol{y}(t)$. The control input is defined as

$$\boldsymbol{u}(t) = -\boldsymbol{K}_O\left(\hat{\boldsymbol{x}}(t) - \boldsymbol{r}(t)\right), \tag{2.20}$$

where $\boldsymbol{K}_O \in \mathbb{R}^{2\times 4}$ is the feedback gain matrix, which can be computed as though all state variables are measurable, depending only on the $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices. With this control input, the observer is given by

$$\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{L}(\boldsymbol{y}(t) - \boldsymbol{C}\hat{\boldsymbol{x}}(t)), \tag{2.21}$$

where $\boldsymbol{L} \in \mathbb{R}^{4\times 2}$, the observer gain matrix, must be defined such that $\boldsymbol{A}-\boldsymbol{L}\boldsymbol{C}$ is a Hurwitz matrix as explained by Åström and Murray (2010). Since $\boldsymbol{D}$ was identified as a matrix of zeros, it is not included in the equations, which is a common technique in observer design.

By taking the error between the states and their estimates as $\tilde{\boldsymbol{x}}(t) = \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)$, the error dynamic equation for the closed-loop system follows from equations 2.19 and 2.21:

$$\dot{\tilde{\boldsymbol{x}}}(t) = (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\tilde{\boldsymbol{x}}(t). \tag{2.22}$$

which leads to $\tilde{\boldsymbol{x}}(t) \to 0$, when $t \to \infty$ when $\boldsymbol{A}-\boldsymbol{LC}$ is a Hurwitz matrix.

The following $\boldsymbol{K}_O$ and $\boldsymbol{L}$ matrices were computed for the 25 mm extension:

$$\boldsymbol{K}_O = \begin{bmatrix} -3.8929 & 2.3760 & -0.1061 & 0.2888 \\ 3.6672 & -0.9670 & 0.0949 & -0.2706 \end{bmatrix},$$

$$\boldsymbol{L} = \begin{bmatrix} 18.5160 & -21.9676 \\ 0.1852 & -7.7968 \\ -0.0051 & 0.0232 \\ -0.0649 & 0.0042 \end{bmatrix}.$$

2.2.8 H_∞-Optimal Output-Feedback Controller

The H_∞-optimal controller is designed using Linear Matrix Inequality (LMI) methods presented by Duan and Yu (2013); Boyd *et al.* (1994); MATLAB's YALMIP toolbox of Lofberg (2004) is then used to solve the optimization problem numerically. The interconnected system $S(\boldsymbol{K}_{H_\infty}, \boldsymbol{G}_o)$ of the optimal gain matrix $\boldsymbol{K}_{H_\infty} \in \mathbb{R}^{2\times 2}$ and the open-loop system $\boldsymbol{G}_o(s)$, with external input defined as $\boldsymbol{w} = [\boldsymbol{r}^T \ \boldsymbol{d}^T \ \boldsymbol{n}^T]^T \in \mathbb{R}^{6\times 1}$ and external output $\boldsymbol{z} = \boldsymbol{r} - \boldsymbol{y}$, represents the closed-loop system with the H_∞ gain:

$$\|\boldsymbol{z}\|_{L_2} \ \leq \ \|S(\boldsymbol{K}_{H_\infty}, \boldsymbol{G}_o)\|_{H_\infty} \|\boldsymbol{w}\|_{L_2}. \tag{2.23}$$

The optimal gain matrix $\boldsymbol{K}_{H_\infty}$ is obtained as the solution to an optimization problem that minimizes the effect of the external input ($\boldsymbol{w}$) on the external output ($\boldsymbol{z}$). We can prove that the H_∞ gain is bounded using the bounded-real lemma from Boyd *et al.* (1994):

Lemma: *Suppose that*

$$\boldsymbol{G}(s) = \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{C} & \boldsymbol{D} \end{array}\right].$$

Then, the following statements are equivalent:

1. $\|\boldsymbol{G}(s)\|_{H_\infty} \leq \gamma$

2. There exists a $\boldsymbol{P} > 0$ such that

$$\begin{bmatrix} \boldsymbol{A}^T\boldsymbol{P}+\boldsymbol{P}\boldsymbol{A} & \boldsymbol{P}\boldsymbol{B} \\ \boldsymbol{B}^T\boldsymbol{P} & -\gamma\boldsymbol{I} \end{bmatrix} + \frac{1}{\gamma}\begin{bmatrix} \boldsymbol{C}^T \\ \boldsymbol{D}^T \end{bmatrix}\begin{bmatrix} \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} < 0$$

The proof that statement 1 implies 2 requires the Hamiltonian, and the proof that statement 2 implies 1 that uses the global stability conditions of the Lyapunov function which is defined as $V(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}$, in which $\boldsymbol{x}^T = [\boldsymbol{x}(t) \;\; \boldsymbol{u}(t)]$ from (2.19).

The control law is designed in the the output-feedback tracking structure:

$$\boldsymbol{u}(t) = -\boldsymbol{K}_{H_\infty}(\boldsymbol{r}(t) - \boldsymbol{y}(t) - \boldsymbol{n}(t)). \tag{2.24}$$

The optimization problem is then solved by using the following theorem from Duan and Yu (2013) to compute the $\boldsymbol{K}_{H_\infty}$ matrix for the closed-loop output-feedback tracking problem. The gain matrix for the 25 mm extension was computed as

$$\boldsymbol{K}_{H_\infty} = \begin{bmatrix} -1.7371 & 2.9015 \\ -0.3775 & -2.4158 \end{bmatrix}.$$

In order to minimize the effect of the external input on the external output while searching for an H_∞-optimal controller matrix $\boldsymbol{K}_{H_\infty}$, a new state-space representation for the tracking framework is given by

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \boldsymbol{z} \\ \boldsymbol{y} \end{bmatrix} = \left[\begin{array}{c|cc} \boldsymbol{A} & \boldsymbol{B}_1 & \boldsymbol{B}_2 \\ \hline \boldsymbol{C}_1 & \boldsymbol{D}_{11} & \boldsymbol{D}_{12} \\ \boldsymbol{C}_2 & \boldsymbol{D}_{21} & \boldsymbol{D}_{22} \end{array}\right] \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{w} \\ \boldsymbol{u} \end{bmatrix}, \tag{2.25}$$

$$\begin{aligned}
&\boldsymbol{A} = \boldsymbol{A}_{4\times 4},\\
&\boldsymbol{B}_1 = \begin{bmatrix} \mathbf{0}_{4\times 2} & \boldsymbol{B}_{4\times 2} & \mathbf{0}_{4\times 2} \end{bmatrix},\ \boldsymbol{B}_2 = \boldsymbol{B}_{4\times 2},\\
&\boldsymbol{C}_1 = -\boldsymbol{C}_{2\times 4},\ \boldsymbol{C}_2 = \boldsymbol{C}_{2\times 4},\\
&\boldsymbol{D}_{11} = \begin{bmatrix} \boldsymbol{I}_{2\times 2} & -\boldsymbol{D}_{2\times 2} & \mathbf{0}_{2\times 2} \end{bmatrix},\ \boldsymbol{D}_{12} = -\boldsymbol{D}_{2\times 2},\\
&\boldsymbol{D}_{21} = \begin{bmatrix} \mathbf{0}_{2\times 2} & \boldsymbol{D}_{2\times 2} & \mathbf{0}_{2\times 2} \end{bmatrix},\ \boldsymbol{D}_{22} = \boldsymbol{D}_{2\times 2},
\end{aligned} \tag{2.26}$$

Theorem: *Let $\boldsymbol{N}_o$ and $\boldsymbol{N}_c$ be full-rank matrices whose images satisfy*

$$\begin{aligned}
Im\ \boldsymbol{N}_o &= Ker\ \begin{bmatrix} \boldsymbol{C}_2 & \boldsymbol{D}_{21} \end{bmatrix},\\
Im\ \boldsymbol{N}_c &= Ker\ \begin{bmatrix} \boldsymbol{B}_2{}^T & \boldsymbol{D}_{12}{}^T \end{bmatrix}.
\end{aligned}$$

The following statements are then equivalent.

1. There exists a $\boldsymbol{K}$ such that $\|S(\boldsymbol{K}, \boldsymbol{G}_o)\|_{H_\infty} < 1$.
2. There exist $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ such that

$$\begin{bmatrix} \boldsymbol{P}_1 & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{P}_2 \end{bmatrix} > 0,$$

$$\begin{bmatrix} \boldsymbol{N}_o & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{P}_1 + \boldsymbol{P}_1\boldsymbol{A} & *^T & *^T \\ \boldsymbol{B}_1^T\boldsymbol{P}_1 & -\boldsymbol{I} & *^T \\ \boldsymbol{C}_1 & \boldsymbol{D}_{11} & -\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{N}_o & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} < 0,$$

$$\begin{bmatrix} \boldsymbol{N}_c & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{P}_2 + \boldsymbol{P}_2\boldsymbol{A} & *^T & *^T \\ \boldsymbol{C}_1\boldsymbol{P}_2 & -\boldsymbol{I} & *^T \\ \boldsymbol{B}_1^T & \boldsymbol{D}_{11}^T & -\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{N}_c & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} < 0.$$

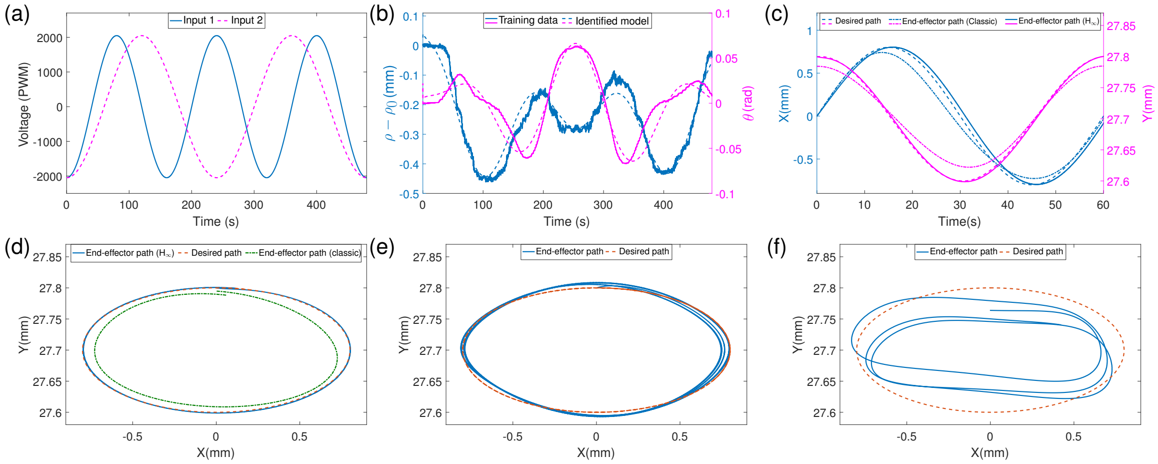


Figure 2.9: Simulation results. (a) 12-bit PWM waveform applied to the SVAs during training. (b) System output and model fitted from training. (c) The x and y coordinates of the manipulator tip over time for the H_∞-optimal and Observer-based controllers, tracking an ellipse. (d) The x-y trajectory of the tip during the simulation in (c). (e) H_∞-optimal controller with noise and disturbance. (f) Observer-based controller with noise and disturbance.

In the matrix inequalities above, "$*$" is used to represent symmetric elements of the matrix. $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are positive definite matrices that are used to construct the Lyapunov function V. Duan and Yu (2013) show that the optimal controller satisfies the Schur stabilizability and H_∞ gain boundedness conditions.

2.2.9 Results

In this section, we study the performance of H_∞ and observer controllers for trajectory tracking. An elliptical reference trajectory is used, defined by

$$\boldsymbol{r}(t) = \begin{bmatrix} \alpha \sin\left(\frac{2\pi}{60}t\right) & \beta + \gamma \cos\left(\frac{2\pi}{60}t\right) \end{bmatrix}^T. \tag{2.27}$$

Table 2.2: N(MAE) of Controller Performance in Simulation (in mm).

Controller	Noise & disturbance	x MAE	y MAE	$x-y$ MAE	$x-y$ NMAE (%)
H_∞	No	0.045	0.006	0.052	3.2
Observer	No	0.065	0.008	0.079	4.9
H_∞	Yes	0.047	0.005	0.055	3.4
Observer	Yes	0.078	0.009	0.096	5.9

where $\alpha = 0.8$, $\beta = 27.7$, and $\gamma = 0.1$ for the 25 mm extension, and $\alpha = 0.6$, $\beta = 11.5$, and $\gamma = 0.3$ for the 9 mm extension, to ensure that each path lies in the workspace of its corresponding manipulator (see Fig. 2.7). The manipulator's tracking performance degraded at frequencies of higher than one cycle per minute.

2.2.10 Comparison of Controllers in Simulation

The performance of the two controllers is first compared in simulation in the presence of the following disturbance and noise signals:

$$\boldsymbol{d}(t) = \begin{bmatrix} 0.00015\sin\left(\frac{3\pi}{60}t\right) & 0.00045\sin\left(\frac{2\pi}{60}t\right) \end{bmatrix}^T, \tag{2.28}$$

$$\boldsymbol{n}(t) = \begin{bmatrix} 0.3\sin\left(\frac{\pi}{60}t\right) & 0.3\sin\left(\frac{0.5\pi}{60}t\right) \end{bmatrix}^T. \tag{2.29}$$

The manipulator with the 25 mm extension was simulated in MATLAB Simulink, using the output-feedback tracking framework depicted in Fig. 2.8 and the identified model and controller values designed in the previous section. Figures 2.9c and 2.9d plot the x and y coordinates and the trajectory of the manipulator tip over time for one cycle (60 s), given an elliptical reference trajectory, from each controller. To observe the effect of adding noise and disturbance in simulation, the sinusoidal functions of $\boldsymbol{n}$ and $\boldsymbol{d}$ were input to the 25 mm manipulator. The tracking trajectories of the manipulator tip produced by each controller

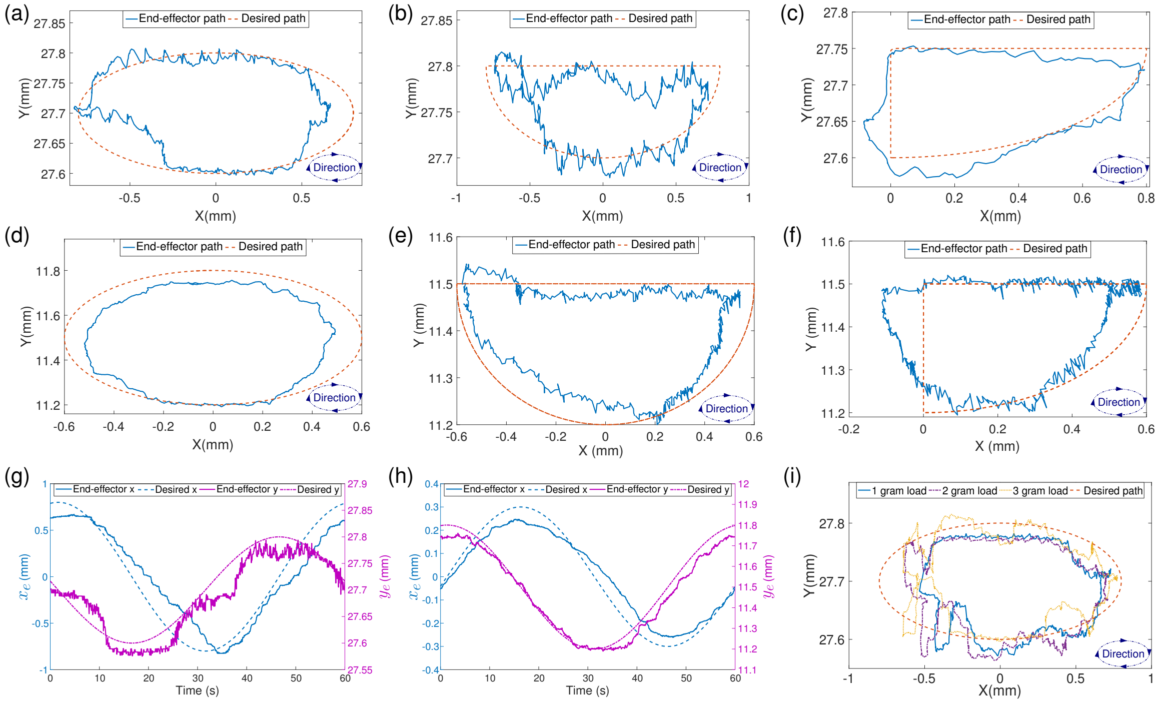


Figure 2.10: Tracking reference and experimental trajectories of the manipulator tip in Cartesian coordinates. 25 mm manipulator tracking: (a) an elliptical trajectory; (b) a half-ellipse; (c) a quarter-ellipse. 9 mm manipulator tracking: (d) an elliptical trajectory; (e) a half-ellipse; (f) a quarter-ellipse. (g) 25 mm manipulator tracking an elliptical trajectory: x, y coordinates over time separately. (h) 9 mm manipulator tracking an elliptical trajectory: x, y coordinates over time separately. (i) 25 mm manipulator tracking an elliptical trajectory under 1 g, 2 g, and 3 g.

are compared in Figs. 2.9e and 2.9f, separately. Although the simulations are performed across three cycles, only one cycle is shown in the figures and used in the error comparison for clarity. The tracking error for each case is reported in Table 2.2. The NMAE values were computed by dividing the mean absolute error (MAE) over their corresponding range. All the values are relatively low, under 10%, indicating accurate tracking.

2.2.11 Experimental Validation of H_∞-Optimal Controller

Since the H_∞-optimal controller exhibited higher tracking accuracy in simulation both with and without disturbance and noise, it was selected for experimental implementation. Using the designed control gain for H_∞, we have implemented the output-feedback tracking framework depicted in Fig. 2.8b on the hydrogel-based manipulator (Fig. 2.6). Half-ellipse and quarter-ellipse paths were also used as reference trajectories. Sources of noise in the experiment arise in the testing environment and vision-based feedback. Disturbances include modeling and manufacturing errors. The MAE and NMAE values are reported for one cycle per trajectory in Table 2.3, though three repeating cycles per trajectory were collected.

Figure 2.10a compares the trajectory of a manipulator with the 25 mm extension driven by the controller (2.24) along an elliptical reference trajectory. Controller performance was evaluated using a half-ellipse and quarter-ellipse reference trajectory as well, to verify the ability of the controlled system to track straight lines and sharp turns (Figs. 2.10b and 2.10c). Figures 2.10d, 2.10e ,and 2.10f show the controlled position of the 9 mm extension's tip using the same reference trajectories. Figures 2.10g and 2.10h illustrate the time evolution of the x and y coordinates separately for the two extensions.

In order to further characterize our system's actuation capabilities, the manipulator's trajectory-tracking performance under load was studied, as shown in Fig. 2.10i. Loads (stainless steel nuts) weighing 1 g, 2 g, and 3 g were placed on the 25 mm extension, as shown in Fig. 2.7b. The manipulator was commanded to follow the same elliptical trajectory as in the unloaded case. The results show that the addition of a weight of up to 3 g increases the trajectory tracking NMAE from 8.1$\%$ to 10.2$\%$ (see Table 2.3). Despite the increase in error, each actuator is still able to function under a load as large as 12.5 times its own weight (0.12 g). As shown in Table 2.3, the experimental NMAE values are higher

Table 2.3: (N)MAE of H_∞ Controller Performance in Experiment.

d	Reference	Load	x	y	$x-y$	NMAE
(mm)	trajectory	(g)	(mm)	(mm)	(mm)	$\%$
25	Ellipse	-	0.123	0.042	0.131	8.1
25	Half Ellipse	-	0.119	0.023	0.123	7.6
25	Quarter Ellipse	-	0.112	0.026	0.119	7.4
9	Ellipse	-	0.088	0.033	0.099	7.4
9	Half Ellipse	-	0.129	0.058	0.132	9.8
9	Quarter Ellipse	-	0.161	0.061	0.171	12.8
25	Ellipse	1	0.140	0.022	0.144	8.9
25	Ellipse	2	0.162	0.021	0.162	10.1
25	Ellipse	3	0.164	0.022	0.164	10.2

than the simulation values, but remain below 15%.

2.2.12 Payload Transport Application

Inspired by the way starfish transport food using their tube feet (Figs. 2.11b and 2.11c) studied by Kerkut (1953); Pentreath (1970), we configured an array of four 9 mm manipulators, as shown in Fig. 2.11e, and applied the proposed H_∞-optimal controller in (2.24) and Fig. 2.8b to each manipulator in order to transport a payload across their tips. The payload being transported is a clear acrylic plate. The manipulators are commanded to track reference trajectories as depicted in Fig. 2.11a, with phase shifts between adjacent manipulators. The payload moves to the right as the manipulators complete repeated cycles of the reference trajectories ("gait cycles"), as shown in Figs. 2.11d and 2.11e. The data from Fig. 2.11d on the duration of one gait cycle and the payload displacement in each

Table 2.4: Payload Displacement ΔX with Different Reference Trajectories for The Manipulators.

Reference trajectory	Payload weight + load (g)	Time for one gait cycle (s)	ΔX after five gait cycles (mm)
Ellipse	2.7	60	5.66
Half-ellipse	2.7	50	4.55
Quarter-ellipse	2.7	40	7.10
Ellipse	2.7+1	60	4.75
Ellipse	2.7+2	60	4.84
Ellipse	2.7+3	60	2.30

tested scenario including the ones with extra added loads on the payload are reported in Table 2.4. The payload's position is recorded but not controlled in this exemplar application, since our goal in this section was to demonstrate a use-case for trajectory tracking control. However, many other platforms and applications are possible, including bio-inspired ones presented in Section 3.2. Through this example, we have demonstrated how trajectory tracking control of systems with soft actuators, when applied to even simple platforms such as this 2-DOF manipulator, may be used to complete complex tasks such as object transport when used in parallel. This type of design can be used to simplify and decouple the control structures in future applications to reduce computational expense.

2.2.13 Discussion

In this section, we addressed a trajectory-tracking problem for a millimeter-scale 2-DOF manipulator with soft hydrogel-based actuators. We defined a linear state-space model of the manipulator and fit the matrices of this model using input-output measurement black-box identification. This state-space representation enables the implementation of a range of controllers on the manipulator; in this section, the performance of an observer-based controller was compared in simulation to that of an H_∞-optimal controller in an output-feedback framework with and without noise and disturbance. We showed experimentally that different versions of the manipulator are able to track various reference trajectories, even under load, using the H_∞-optimal controller.

As directions for future work, the speed of the image processing algorithms can be improved for tracking the manipulator, and the use of the camera for position tracking could be eliminated and instead, embedded sensor feedback could be used to implement the current control approach. This would also enable the application of machine learning techniques to optimize control performance.

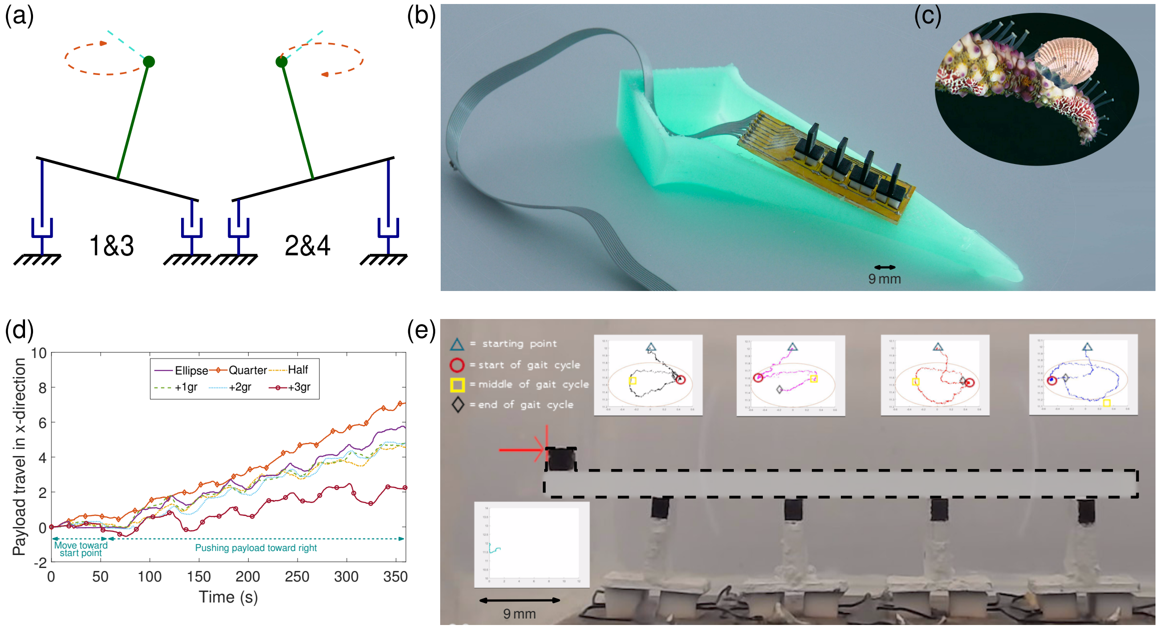


Figure 2.11: Control of four 9 mm manipulators in series for payload transport, in a manner similar to food transport by starfish tube feet. (a) The manipulators, numbered 1 to 4 from left to right, are commanded to first follow the cyan dashed lines from their initial positions to their starting positions on the reference trajectories, and then follow these trajectories, shown as red dashed lines. Manipulators 1 and 3 have a phase shift of 180° compared to manipulators 2 and 4. (b) Illustration of a starfish-inspired robotic platform with four hydrogel-actuated manipulators. (c) Real starfish transporting a clam on its tube feet. (d) Displacement of the payload as a function of time for different reference trajectories and load weights. (e) Array of four manipulators functioning as described in (a) to transport the payload. Image was taken when the manipulators completed the first gait cycle. The payload is a clear flat acrylic plate with a black square on its left side. The positions of the manipulator tips are marked by triangles at their initial locations, circles at the start of the gait cycle, squares at the middle of the gait cycle, and diamonds at the end of the gait cycle.

Chapter 3

DYNAMIC MODELING AND CONTROL OF SOFT CONTINUUM ROBOT ARMS BASED ON THE COSSERAT ROD MODEL

This chapter describes research published by Doroudchi and Berman (2021); Doroudchi *et al.* (2020), and Doroudchi *et al.* (2022), a manuscript in preparation on experimental validation of the configuration tracking controller using a silicone-based multi-segment soft robotic arm.

3.1 Configuration Tracking using Inverse Dynamic Control of the Cosserat Rod Model

Controlling the configuration of a soft continuum robot arm is challenging due to the hyper-redundant kinematics of such robots. We propose a new model-based, inverse dynamic control approach to this problem that is defined on the configuration state variables of the geometrically exact Cosserat rod model. Our approach is capable of controlling a soft continuum robot to track static or time-varying 3D configurations through bending, torsion, shear, and extension deformations. The controller has a decentralized structure, in which the gain matrices can be defined in terms of the physical and material properties of distinct cross-sections of the robot arm. This structure facilitates its implementation on continuum robot arms composed of independently-controllable segments that have local sensing and actuation. The controller is validated with numerical simulations in MATLAB with a hydrogel-based soft robot arm that can produce the four primary types of deformations. The simulated arm successfully tracks these configurations with average normalized root-mean-square errors (NRMSE) below 7% in all cases. To demonstrate the generality of the control approach, its performance is also validated on a larger simulated robot arm made of silicone.

3.1.1 Forward Dynamics of a Cosserat Rod Model

We aim to develop a model of the nonlinear dynamics of an octopus-inspired soft continuum robot arm that accounts for the effects of large deformations due to bending, torsion, shear, and extension. A suitable candidate is the Cosserat rod model, for which three assumptions are required: a sufficiently large length-to-diameter ratio, material incompressability, and linear elasticity, as stated by Antman (2005).

Figure 3.1 depicts a 3-dimensional uniform Cosserat rod in Cartesian coordinates. The length of the rod is denoted by L, the density of the rod by ρ, and the area and second mass moment of inertia tensor of each cross-section by A and $\boldsymbol{J}$, respectively. The position and orientation matrix of each cross-section at arc length s in the global coordinate frame are denoted by ${}^{G}\boldsymbol{p}(t,s) \in \mathbb{R}^3$ and ${}^{G}\boldsymbol{R}(t,s) \in SO(3)$, respectively. From this point on, whenever a variable does not have the global frame annotation G, it means that it is defined with respect to a local coordinate frame that is fixed to the cross-section in which the variable is defined. In the Cosserat dynamics of a rod whose neutral axis is in the z direction, the curvature vector $\boldsymbol{u}(t,s) = [\boldsymbol{u}_x, \boldsymbol{u}_y, \boldsymbol{u}_z]^T$ and the rate of change of position $\boldsymbol{v}(t,s) = [\boldsymbol{v}_x, \boldsymbol{v}_y, \boldsymbol{v}_z]^T$ are directly responsible for deformations of the rod, and we will refer to them as configuration state variables. The vectors $\boldsymbol{u}_x$ and $\boldsymbol{u}_y$ produce bending about the x and y axes, and $\boldsymbol{u}_z$ creates torsion about the z-axis. The vectors $\boldsymbol{v}_x$ and $\boldsymbol{v}_y$ cause shear effects that produce changes in the size of the cross-section, and $\boldsymbol{v}_z$ produces extension along the z-axis. The vectors $\boldsymbol{q}(t,s)$ and $\boldsymbol{w}(t,s)$ define velocity and angular velocity. The internal force and moment are represented by ${}^{G}\boldsymbol{n}(t,s) \in \mathbb{R}^3$ and ${}^{G}\boldsymbol{m}(t,s) \in \mathbb{R}^3$, respectively. They are the force and moment that the material at $\boldsymbol{p}(t, s+ds)$ exerts on the material at $\boldsymbol{p}(t, s-ds)$, for infinitesimal ds.

A set of partial differential equations, differentiated with respect to arc length s and time t, governs the deformation of each cross-section along the elastic Cosserat rod (Fig. 3.1).

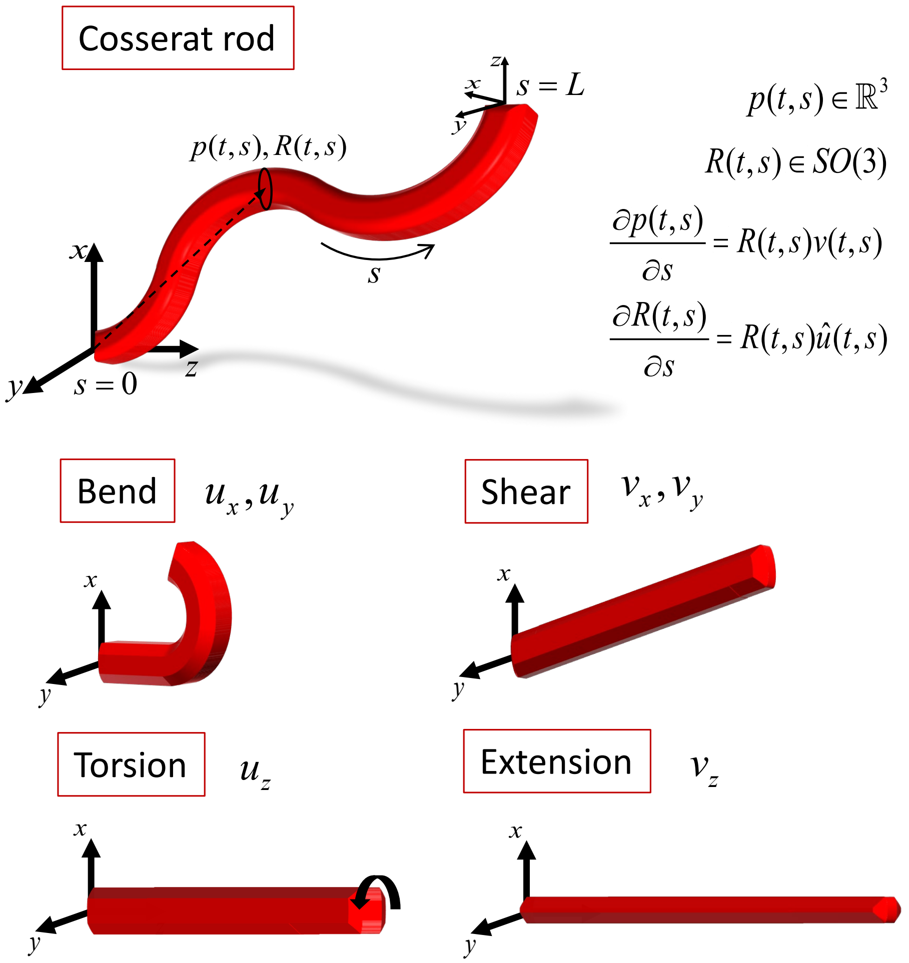

Cosserat rod
$s = L$
x
y
z
$p(t,s), R(t,s)$
s
$s = 0$
$p(t,s) \in \mathbb{R}^3$
$R(t,s) \in SO(3)$
$\frac{\partial p(t,s)}{\partial s} = R(t,s)v(t,s)$
$\frac{\partial R(t,s)}{\partial s} = R(t,s)\hat{u}(t,s)$
Bend
u_x, u_y
Shear
v_x, v_y
Torsion
u_z
Extension
v_z

Figure 3.1: Schematic of a Cosserat rod. The equations for the 3-dimensional rod configuration describe the relationships between the curvature vector ($\boldsymbol{u}$), the rate of change of position with respect to arc length ($\boldsymbol{v}$), and the resulting bending, torsion, shear, and extension deformations.

The spatial derivatives of the state variables are calculated at each cross-section of the rod. The internal force and moment evolve according to the equations:

$$\begin{aligned} {}^G\boldsymbol{n}_s &= {}^G\boldsymbol{R}\rho A(\widehat{\boldsymbol{w}}\boldsymbol{q} + \boldsymbol{q}_t) - {}^G\boldsymbol{f}, \\ {}^G\boldsymbol{m}_s &= {}^G\boldsymbol{R}\rho(\widehat{\boldsymbol{w}}\boldsymbol{J}\boldsymbol{w} + \boldsymbol{J}\boldsymbol{w}_t) - {}^G\widehat{\boldsymbol{p}}_s{}^G\boldsymbol{n} - {}^G\boldsymbol{l}, \end{aligned} \tag{3.1}$$

and the kinematic variables evolve according to:

$$\begin{aligned} {}^G\boldsymbol{p}_s &= {}^G\boldsymbol{R}\boldsymbol{v}\ , {}^G\boldsymbol{p}_t = \boldsymbol{R}\boldsymbol{q}, \\ {}^G\boldsymbol{R}_s &= {}^G\boldsymbol{R}\widehat{\boldsymbol{u}}\ , {}^G\boldsymbol{R}_t = \boldsymbol{R}\widehat{\boldsymbol{w}}, \\ \boldsymbol{q}_s &= \boldsymbol{v}_t - \widehat{\boldsymbol{u}}\boldsymbol{q} + \widehat{\boldsymbol{w}}\boldsymbol{v}, \\ \boldsymbol{w}_s &= \boldsymbol{u}_t - \widehat{\boldsymbol{u}}\boldsymbol{w}, \end{aligned} \tag{3.2}$$

where $\widehat{(\cdot)}$ is the cross product matrix of a vector. Stoer and Bulirsch (2013) compute the time derivatives using the Backward Differentiation Formula (BDF) as follows:

$$\begin{aligned} \boldsymbol{v}_t &= c_0\boldsymbol{v}^i + \boldsymbol{v}_h,\ \ \boldsymbol{v}_h = c_1\boldsymbol{v}^{i-1} + c_2\boldsymbol{v}^{i-2}, \\ \boldsymbol{u}_t &= c_0\boldsymbol{u}^i + \boldsymbol{u}_h,\ \ \boldsymbol{u}_h = c_1\boldsymbol{u}^{i-1} + c_2\boldsymbol{u}^{i-2}, \\ \boldsymbol{q}_t &= c_0\boldsymbol{q}^i + \boldsymbol{q}_h,\ \ \boldsymbol{q}_h = c_1\boldsymbol{q}^{i-1} + c_2\boldsymbol{q}^{i-2}, \\ \boldsymbol{w}_t &= c_0\boldsymbol{w}^i + \boldsymbol{w}_h,\ \ \boldsymbol{w}_h = c_1\boldsymbol{w}^{i-1} + c_2\boldsymbol{w}^{i-2}, \end{aligned} \tag{3.3}$$

in which $c_0 = 1.5/dt$, $c_1 = -2/dt$, and $c_2 = 0.5/dt$ are the implicit difference coefficients. The elements $(\cdot)^{(i)}$ are the values of the corresponding variables at time step i, and the history elements, $(\cdot)_h$, are the values of the corresponding variables at the two previous time steps, $i-1$ and $i-2$.

The equations of an elastic Cosserat rod are written for its central backbone, and any forces and moments that are applied to this backbone are modeled as external forces and moments, denoted by ${}^G\boldsymbol{f}$ and ${}^G\boldsymbol{l}$, respectively. In practice, these forces and moments are the sum of the control inputs applied by the actuators of the continuum robot arm (${}^G\boldsymbol{f}_a, {}^G\boldsymbol{l}_a$), external loads on the arm (${}^G\boldsymbol{f}_l, {}^G\boldsymbol{l}_l$), and environmental effects (${}^G\boldsymbol{f}_e, {}^G\boldsymbol{l}_e$):

$$\begin{aligned} {}^G\boldsymbol{f} &= {}^G\boldsymbol{f}_a + {}^G\boldsymbol{f}_l + {}^G\boldsymbol{f}_e, \\ {}^G\boldsymbol{l} &= {}^G\boldsymbol{l}_a + {}^G\boldsymbol{l}_l + {}^G\boldsymbol{l}_e. \end{aligned} \tag{3.4}$$

Our approach to numerically solving (3.1) is outlined in Algorithm 4. The implicit fourth-order Runge-Kutta (RK4) method is implemented to numerically integrate (3.1) with respect to space, and then the standard shooting method (SSM) is used to generate initial guesses $\boldsymbol{n}_0, \boldsymbol{m}_0$ of $\boldsymbol{n}, \boldsymbol{m}$ at each time step. The following boundary conditions of the fixed end of the rod are known:

$$\begin{aligned} {}^{G}\boldsymbol{p}(t,0) &= \boldsymbol{p}_0,\ {}^{G}\boldsymbol{R}(t,0) = \boldsymbol{R}_0, \\ \boldsymbol{q}(t,0) &= 0,\ \ \boldsymbol{w}(t,0) = 0. \end{aligned} \tag{3.5}$$

The SSM guesses the following unknown boundary conditions of the fixed end of the rod,

$${}^{G}\boldsymbol{n}(t,0) = \boldsymbol{n}_0,\ {}^{G}\boldsymbol{m}(t,0) = \boldsymbol{m}_0, \tag{3.6}$$

while satisfying the known boundary conditions of the free end,

$${}^{G}\boldsymbol{n}(t,L) = \boldsymbol{n}_L = \mathbf{0},\ {}^{G}\boldsymbol{m}(t,L) = \boldsymbol{m}_L = \mathbf{0}. \tag{3.7}$$

This two-point Boundary Value Problem (BVP), which has been reduced to an Initial Value Problem (IVP) as explained by Stoer and Bulirsch (2013); Holsapple (2003), is solved by the SSM guesses at each iteration of Algorithm 4. Then, the guessed values are corrected by the Levenberg-Marquardt nonlinear optimization algorithm. A unique solution to the BVT is ensured by using a sufficiently small time step and using the solution at the previous time step as the initial guess for the current time step as proposed by Till and Rucker (2017). Then, the configuration state variables, $\boldsymbol{v}$ and $\boldsymbol{u}$, are found from the computed ${}^{G}\boldsymbol{n}$ and ${}^{G}\boldsymbol{m}$ as follows:

$$\begin{aligned} \boldsymbol{v} &= (\boldsymbol{K}_{se} + c_0\boldsymbol{B}_{se})^{-1}[{}^{G}\boldsymbol{R}^{T}\,{}^{G}\boldsymbol{n} + \boldsymbol{K}_{se}\boldsymbol{v}^{*} - \boldsymbol{B}_{se}\boldsymbol{v}_h], \\ \boldsymbol{u} &= (\boldsymbol{K}_{bt} + c_0\boldsymbol{B}_{bt})^{-1}[{}^{G}\boldsymbol{R}^{T}\,{}^{G}\boldsymbol{m} + \boldsymbol{K}_{bt}\boldsymbol{u}^{*} - \boldsymbol{B}_{bt}\boldsymbol{u}_h]. \end{aligned} \tag{3.8}$$

The vectors $\boldsymbol{v}^{*}$ and $\boldsymbol{u}^{*}$ are the values of $\boldsymbol{v}$ and $\boldsymbol{u}$ at the undeformed reference shape. The effects of shear and extension are characterized by $\boldsymbol{K}_{se}$ and the effects of bending and torsion by $\boldsymbol{K}_{bt}$, under the assumption that the continuum robot arm is slender, symmetric, homogeneous, and isotropic:

$$\boldsymbol{K}_{se} = \begin{bmatrix} \alpha_c G & 0 & 0 \\ 0 & \alpha_c G & 0 \\ 0 & 0 & E \end{bmatrix} A, \quad \boldsymbol{K}_{bt} = \begin{bmatrix} E & 0 & 0 \\ 0 & E & 0 \\ 0 & 0 & G \end{bmatrix} \boldsymbol{J}. \tag{3.9}$$

In these equations, G and E are the shear modulus and Young's modulus, respectively, and α_c is a constant which is equal to 4/3 for circular cross-sections and 3/2 for rectangular ones. The damping matrices $\boldsymbol{B}_{se} = \tau \boldsymbol{K}_{se}$ and $\boldsymbol{B}_{bt} = \tau \boldsymbol{K}_{bt}$ in Eq. (3.8) can be calculated from vibration tests presented by Linn *et al.* (2013), in which τ is twice the period of vibrations exhibited by the continuum robot arm's tip.

3.1.2 A Decentralized Approach to Inverse Dynamic Control

We design a controller with a decentralized structure that has distinct proportional-derivative (PD) gains at each location along the arc length of the Cosserat rod. For segmented soft continuum robots, this enables the independent control of each segment in a computationally efficient manner. We have previously developed decentralized control approaches for segmented soft robot arms with objectives of vibration dampening using H_∞ state feedback control, presented in Section 2.1, and trajectory tracking using a consensus-based method proposed by Lafmejani *et al.* (2020b). Here, a desired reference configuration is tracked ($\bar{\boldsymbol{v}}$, $\bar{\boldsymbol{u}}$), which may be static or time-varying and results in bending, twisting, shear, extension, or a combination of these deformations in 3D space. In another study by Lafmejani *et al.* (2020a), we developed a kinematic controller for trajectory tracking by the tip of a segmented hyper-redundant robot arm, modeled as a series of Gough-Stewart platforms, that is capable of producing all four deformations considered here. In the Lafmejani *et al.* (2020a) study and this section, we consider robot arms that are comprised of a series of physically connected segments with local sensing and actuation, and both control approaches rely on calculations by a central computational unit. However, we use a kinematic model of the robot demonstrated by Lafmejani *et al.* (2020a) and a Cosserat-based dynamic model in this section, and the robot's segments from Lafmejani *et al.* (2020a) are assumed to exchange information with adjacent segments, which is not assumed here. Moreover, the control approach developed by Lafmejani *et al.* (2020a) is centralized, whereas the approach presented here is decentralized.

The control law is defined to track a reference configuration ($\bar{\boldsymbol{v}}$, $\bar{\boldsymbol{u}}$) and to compensate for external loads and environmental effects. The control inputs below are applied by the actuators of the continuum robot arm to the backbone of the robot:

$$\begin{aligned}
{}^G\boldsymbol{f}_a ={}& {}^G\boldsymbol{R}[\boldsymbol{K}_{m_1}\bar{\boldsymbol{v}}_{tt} + \boldsymbol{K}_{v_1}(\bar{\boldsymbol{v}}_t - \boldsymbol{v}_t) + \boldsymbol{K}_{p_1}(\bar{\boldsymbol{v}} - \boldsymbol{v})] \\
& - {}^G\boldsymbol{f}_l - {}^G\boldsymbol{f}_e, \\
{}^G\boldsymbol{l}_a ={}& {}^G\boldsymbol{R}[\boldsymbol{K}_{m_2}\bar{\boldsymbol{u}}_{tt} + \boldsymbol{K}_{v_2}(\bar{\boldsymbol{u}}_t - \boldsymbol{u}_t) + \boldsymbol{K}_{p_2}(\bar{\boldsymbol{u}} - \boldsymbol{u})] \\
& - {}^G\boldsymbol{l}_l - {}^G\boldsymbol{l}_e,
\end{aligned} \tag{3.10}$$

where $\boldsymbol{K}_{m_1}, \boldsymbol{K}_{m_2}$ are 3×3 diagonal matrices whose diagonal entries are proportional to $\rho A, \rho\boldsymbol{J}$, respectively, and $\boldsymbol{K}_{v_1}, \boldsymbol{K}_{v_2}, \boldsymbol{K}_{p_1}, \boldsymbol{K}_{p_2}$ are gain matrices defined as:

$$\begin{aligned}
\boldsymbol{K}_{v_1} &= v_1\boldsymbol{B}_{se}, \quad \boldsymbol{K}_{p_1} = p_1\boldsymbol{K}_{se}, \\
\boldsymbol{K}_{v_2} &= v_2\boldsymbol{B}_{bt}, \quad \boldsymbol{K}_{p_2} = p_2\boldsymbol{K}_{bt},
\end{aligned} \tag{3.11}$$

where we set the scalar coefficients to $v_1 = p_1 = p_2 = v_2 = 1$. In this way, the controller gains at a specific cross-section of the robot can be defined in terms of the physical properties (A, $\boldsymbol{J}$) of that cross-section, independently of the composition of other cross-sections. This facilitates its implementation on a segmented continuum robot arm with a decentralized control structure.

Algorithm 4 briefly describes our implementation of this configuration tracking controller. First, the desired configuration is defined in terms of the variables $\bar{\boldsymbol{v}}$ and $\bar{\boldsymbol{u}}$ (line 1). We define an *outer loop* that iterates over time steps (lines 2 to 10) and an *inner loop* that iterates over discretized spatial locations (nodes) along the backbone of the robot (lines 4 to 9). In the outer loop, the initial boundary condition values of $\boldsymbol{n}_0$ and $\boldsymbol{m}_0$ are guessed using SSM; in the first iteration, they are set to zero (line 3). By applying RK4 to (${}^G\boldsymbol{n}_{j-1}$,${}^G\boldsymbol{m}_{j-1}$) and the derivatives of the internal force and moment with respect to arc length, their values at the current spatial node on the backbone, j, are computed (line 5). After implementing

Algorithm 4 Configuration tracking controller

```
1:  Given a desired configuration $\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}$
2:  for $i \leftarrow 0$ to $T/dt$ do
3:      $\boldsymbol{n}_0^i, \boldsymbol{m}_0^i \leftarrow$ SSM ($\boldsymbol{n}_L^i = 0, \boldsymbol{m}_L^i = 0$)
4:      for $j \leftarrow 0$ to $L/ds$ do
5:          $\boldsymbol{n}_j^i, \boldsymbol{m}_j^i \leftarrow$ RK4 ($\boldsymbol{n}_{j-1}^i, \boldsymbol{n}_{s,j}^i$) and ($\boldsymbol{m}_{j-1}^i, \boldsymbol{m}_{s,j}^i$)
6:          $\boldsymbol{v}_j^i, \boldsymbol{u}_j^i \leftarrow \boldsymbol{n}_j^i, \boldsymbol{m}_j^i$, Forward dynamics (3.8)
7:          $\boldsymbol{f}_{a,j}^i, \boldsymbol{l}_{a,j}^i \leftarrow$ Control law (3.10)
8:          $\boldsymbol{n}_{s,j}^i, \boldsymbol{m}_{s,j}^i \leftarrow$ Substitute (3.4) in (3.1)
9:      end for
10: end for
11: Note: $(\boldsymbol{n}, \boldsymbol{m}, \boldsymbol{f}, \boldsymbol{l})$ are defined in the global frame, and $(\boldsymbol{v}, \boldsymbol{u})$ in the local frame.
```

the current internal force and moments in (3.8), the values for $\boldsymbol{v}_j$ and $\boldsymbol{u}_j$ are found (line 6). Then, using the error between the configuration variables and their desired values, the control law calculates the force and moment (${}^G\boldsymbol{f}_a$,${}^G\boldsymbol{l}_a$) that the actuators must apply to the corresponding backbone section (line 7), and then ${}^G\boldsymbol{n}_s$ and ${}^G\boldsymbol{m}_s$ are found for the next spatial node on the backbone (line 8). To demonstrate that the configuration tracking error converges to zero under this controller, first the closed-loop system dynamics are derived. By rearranging the equations in (3.1), we obtain:

$$\begin{aligned} {}^G\boldsymbol{f} &= {}^G\boldsymbol{R}\rho A(\widehat{\boldsymbol{w}}\boldsymbol{q} + \boldsymbol{q}_t) - {}^G\boldsymbol{n}_s, \\ {}^G\boldsymbol{l} &= {}^G\boldsymbol{R}\rho(\widehat{\boldsymbol{w}}\boldsymbol{J}\boldsymbol{w} + \boldsymbol{J}\boldsymbol{w}_t) - {}^G\widehat{\boldsymbol{p}}_s {}^G\boldsymbol{n} - {}^G\boldsymbol{m}_s. \end{aligned} \tag{3.12}$$

Then, by substituting in the sum of external forces and moments from (3.4) and keeping the actuator control inputs on the left-hand side, we obtain:

$$\begin{aligned} {}^G\boldsymbol{f}_a &= {}^G\boldsymbol{R}\rho A(\widehat{\boldsymbol{w}}\boldsymbol{q} + \boldsymbol{q}_t) - {}^G\boldsymbol{n}_s - {}^G\boldsymbol{f}_l - {}^G\boldsymbol{f}_e, \\ {}^G\boldsymbol{l}_a &= {}^G\boldsymbol{R}\rho(\widehat{\boldsymbol{w}}\boldsymbol{J}\boldsymbol{w} + \boldsymbol{J}\boldsymbol{w}_t) - {}^G\widehat{\boldsymbol{p}}_s {}^G\boldsymbol{n} - {}^G\boldsymbol{m}_s - {}^G\boldsymbol{l}_l - {}^G\boldsymbol{l}_e. \end{aligned} \tag{3.13}$$

Replacing ${}^{G}\boldsymbol{f}_a$ and ${}^{G}\boldsymbol{l}_a$ in (3.13) with the control law (3.10), the external loads and environmental forces and moments from both sides of the equations cancel out:

$$\begin{aligned}
&{}^{G}\boldsymbol{R}[\boldsymbol{K}_{m_1}\bar{\boldsymbol{v}}_{tt} + \boldsymbol{K}_{v_1}(\bar{\boldsymbol{v}}_t - \boldsymbol{v}_t) + \boldsymbol{K}_{p_1}(\bar{\boldsymbol{v}} - \boldsymbol{v})] \\
&= {}^{G}\boldsymbol{R}\rho A(\widehat{\boldsymbol{w}}\boldsymbol{q} + \boldsymbol{q}_t) - {}^{G}\boldsymbol{n}_s, \\
&{}^{G}\boldsymbol{R}[\boldsymbol{K}_{m_2}\bar{\boldsymbol{u}}_{tt} + \boldsymbol{K}_{v_2}(\bar{\boldsymbol{u}}_t - \boldsymbol{u}_t) + \boldsymbol{K}_{p_2}(\bar{\boldsymbol{u}} - \boldsymbol{u})] \\
&= {}^{G}\boldsymbol{R}\rho(\widehat{\boldsymbol{w}}\boldsymbol{J}\boldsymbol{w} + \boldsymbol{J}\boldsymbol{w}_t) - {}^{G}\widehat{\boldsymbol{p}}_s{}^{G}\boldsymbol{n} - {}^{G}\boldsymbol{m}_s.
\end{aligned} \tag{3.14}$$

The right-hand sides of these equations are the sums of the internal forces and moments with respect to the arc length. By defining $\boldsymbol{n'}_s$ and $\boldsymbol{m'}_s$ as the following expressions,

$$\begin{aligned}
{}^{G}\boldsymbol{n'}_s &= {}^{G}\boldsymbol{R}\rho A(\widehat{\boldsymbol{w}}\boldsymbol{q} + \boldsymbol{q}_t) - {}^{G}\boldsymbol{n}_s, \\
{}^{G}\boldsymbol{m'}_s &= {}^{G}\boldsymbol{R}\rho(\widehat{\boldsymbol{w}}\boldsymbol{J}\boldsymbol{w} + \boldsymbol{J}\boldsymbol{w}_t) - {}^{G}\widehat{\boldsymbol{p}}_s{}^{G}\boldsymbol{n} - {}^{G}\boldsymbol{m}_s,
\end{aligned} \tag{3.15}$$

and rewriting them in terms of the second time derivatives of the configuration state variables,

$$\begin{aligned}
{}^{G}\boldsymbol{n'}_s &= {}^{G}\boldsymbol{R}\boldsymbol{K}_{m_1}\boldsymbol{v}_{tt}, \\
{}^{G}\boldsymbol{m'}_s &= {}^{G}\boldsymbol{R}\boldsymbol{K}_{m_2}\boldsymbol{u}_{tt},
\end{aligned} \tag{3.16}$$

the closed-loop configuration dynamics of the robot can be expressed in the following form:

$$\begin{aligned}
\boldsymbol{K}_{m_1}\bar{\boldsymbol{v}}_{tt} + \boldsymbol{K}_{v_1}(\bar{\boldsymbol{v}}_t - \boldsymbol{v}_t) + \boldsymbol{K}_{p_1}(\bar{\boldsymbol{v}} - \boldsymbol{v}) &= \boldsymbol{K}_{m_1}\boldsymbol{v}_{tt}, \\
\boldsymbol{K}_{m_2}\bar{\boldsymbol{u}}_{tt} + \boldsymbol{K}_{v_2}(\bar{\boldsymbol{u}}_t - \boldsymbol{u}_t) + \boldsymbol{K}_{p_2}(\bar{\boldsymbol{u}} - \boldsymbol{u}) &= \boldsymbol{K}_{m_2}\boldsymbol{u}_{tt}.
\end{aligned} \tag{3.17}$$

Defining the error vector $\boldsymbol{e}(t) = (\bar{\boldsymbol{v}} - \boldsymbol{v}, \bar{\boldsymbol{u}} - \boldsymbol{u})^T$ and writing (3.17) in terms of this error, the closed-loop system dynamics take the form of standard homogeneous second-order differential equations:

$$\boldsymbol{e}_{tt} + \boldsymbol{K'}_v\boldsymbol{e}_t + \boldsymbol{K'}_p\boldsymbol{e} = \boldsymbol{0}, \tag{3.18}$$

where the matrices $\boldsymbol{K'}_v$ and $\boldsymbol{K'}_p$ are defined as

$$\begin{aligned}\boldsymbol{K'}_v &= \begin{bmatrix} \boldsymbol{K}_{v_1} \oslash \boldsymbol{K}_{m_1} & 0 \\ 0 & \boldsymbol{K}_{v_2} \oslash \boldsymbol{K}_{m_2} \end{bmatrix}, \\ \boldsymbol{K'}_p &= \begin{bmatrix} \boldsymbol{K}_{p_1} \oslash \boldsymbol{K}_{m_1} & 0 \\ 0 & \boldsymbol{K}_{p_2} \oslash \boldsymbol{K}_{m_2} \end{bmatrix},\end{aligned} \tag{3.19}$$

in which $\oslash$ denotes element-wise division of matrices (Hadamard division). The matrices $\boldsymbol{K'}_v$ and $\boldsymbol{K'}_p$ are symmetric and positive definite. To show that $\boldsymbol{e}(t) \to \boldsymbol{0}$ as $t \to \infty$, the following positive definite quadratic Lyapunov function is chosen,

$$V = \frac{1}{2}\boldsymbol{e}_t^T \boldsymbol{e}_t + \frac{1}{2}\boldsymbol{e}^T \boldsymbol{K'}_p \boldsymbol{e}, \tag{3.20}$$

which has the following time derivative:

$$\begin{aligned} V_t &= \frac{1}{2}\boldsymbol{e}_{tt}^T \boldsymbol{e}_t + \frac{1}{2}\boldsymbol{e}_t^T \boldsymbol{e}_{tt} + \frac{1}{2}\boldsymbol{e}_t^T \boldsymbol{K'}_p \boldsymbol{e} + \frac{1}{2}\boldsymbol{e}^T \boldsymbol{K'}_p \boldsymbol{e}_t \\ &= \frac{1}{2}(\boldsymbol{e}_{tt}^T + \boldsymbol{e}^T \boldsymbol{K'}_p)\boldsymbol{e}_t + \frac{1}{2}\boldsymbol{e}_t^T(\boldsymbol{e}_{tt} + \boldsymbol{K'}_p \boldsymbol{e}) \\ &= \frac{1}{2}(-\boldsymbol{e}_t^T \boldsymbol{K'}_v)\boldsymbol{e}_t + \frac{1}{2}\boldsymbol{e}_t^T(-\boldsymbol{K'}_v \boldsymbol{e}_t) = -\boldsymbol{e}_t^T \boldsymbol{K'}_v \boldsymbol{e}_t. \end{aligned} \tag{3.21}$$

Since $\boldsymbol{K'}_v$ is positive definite, V_t is a negative definite function. By applying Lyapunov's direct method to the closed-loop system dynamics, we can prove that $\boldsymbol{e}(t) \to \boldsymbol{0}$ as $t \to \infty$ and the system is globally asymptotically stable (Khalil and Grizzle, 2002).

3.1.3 Simulation Results

In this section, the performance of the decentralized configuration tracking controller is validated with numerical simulations of a Cosserat rod model of the hydrogel-based segmented continuum robot arm in Section 3.2. The simulated robot must achieve specified reference configurations through bending, torsion, shear, and extension. In all simulations, the relatively low values of the average normalized root-mean-square errors (NRMSEs)

Table 3.1: Parameters in the Numerical Simulations.

Param.	Description	Value	
ds	Spatial discretization	L/N	m
dt	Time step	0.1	s
N	Number of spatial nodes	80	-
N_v	Number of virtual segments	8	-
r_p	Undeformed radius of proximal segment	3	mm
r_d	Undeformed radius of distal segment	0.5	mm
L	Undeformed length of the hydrogel robot arm	45	mm
ρ	Density of hydrogel actuator (Khodambashi *et al.*, 2019)	1300	kg/m^3
E	Young's modulus of hydrogel actuator	0.6	MPa
ρ_S	Density of silicone	2330	kg/m^3
E_S	Young's modulus of silicone	188	GPa

between the simulated and reference configurations over all spatial nodes (discretized locations along the robot's backbone) demonstrate effective tracking performance by the controller. These values are stated in figure captions in the following subsections.

As in our work on decentralized control of soft segmented continuum robots presented in Section 2.1 and by Lafmejani *et al.* (2020b), we assume that each segment of the robot has local sensing and actuation capabilities. For each test case, the simulated continuum robot arm is slender and symmetric about the z-axis, with isotropic material properties and the parameters listed in Table 3.1.

Although the robot is simulated with circular cross-sections, our control approach can be applied to continuum robots with other cross-section geometries, as long as they satisfy the assumptions required for using the Cosserat rod model. Both uniform and tapered robot arms were simulated; the cross-section of the uniform arm has a constant radius r_p, while the radius of the cross-section of the tapered arm decreases from r_p at the proximal segment to r_d at the distal (tip) segment.

Since hydrogel has a slow response time, in practice, the control input frequencies must be low enough to give the material enough time to complete its heating and cooling phases. To demonstrate the applicability of our controller to other types of materials, we also simulated it on a robot arm composed of silicone, which has a faster response than hydrogel, with the parameters given in Table 3.1. The dimensions L, r_p, and r_d of the simulated silicone robot arm were each 10 times longer than the corresponding dimensions of the hydrogel-based arm. The reference control inputs for the silicone arm were the same as the inputs defined for the hydrogel-based arm, except with amplitudes 10 times larger and, in the case of controlled shear deformations, a frequency 10 times smaller.

For the arm made of hydrogel, which can only deform underwater, the applied force from the environment was defined as the gravitational force acting on the arm,

$$^{G}\boldsymbol{f}_e = (\rho - \rho_w)A\boldsymbol{g} - {}^{G}\boldsymbol{R}\boldsymbol{C}\boldsymbol{q} \odot |\boldsymbol{q}|, \tag{3.22}$$

where ρ is the density of the hydrogel, ρ_w is the density of the surrounding water, $\boldsymbol{g}$ is gravitational acceleration, and $\boldsymbol{C}$ is the damping coefficient matrix of the arm, which is used to define the square-law drag force on the arm due to water resistance ($\odot$ denotes element-wise multiplication). The remaining forces and moments in (3.4) due to external loads and environmental effects were set to zero. For the arm made of silicone, which was assumed to operate in air, the density ρ in (3.22) was set to ρ_S, the density of silicone, and the drag force term was set to zero.

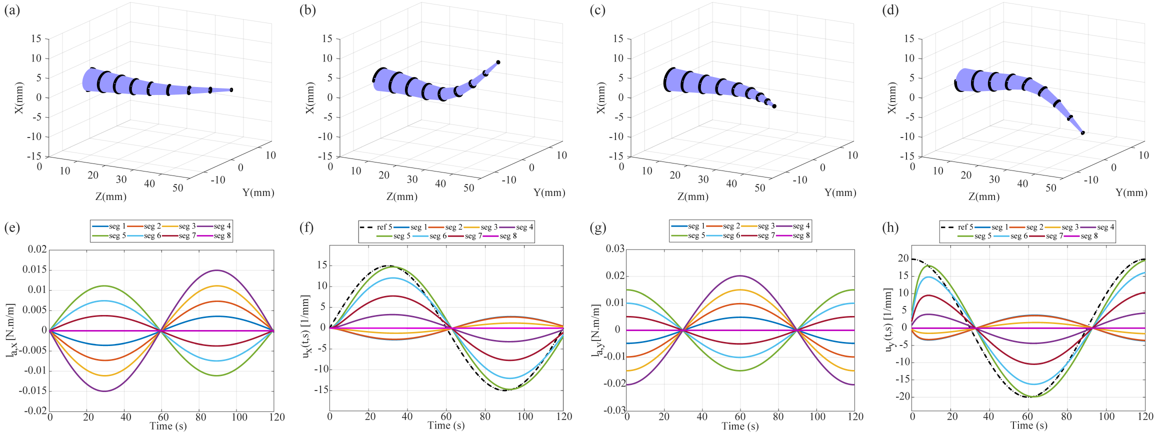


Figure 3.2: Tracking performance for a reference configuration that requires bending deformations. (a)-(d) Simulation snapshots at times $t = 30\,s$, $60\,s$, $90\,s$, and $120\,s$, respectively. (e),(g) Applied moments about the x and y axes, respectively. (f),(h) Curvature components $\boldsymbol{u}_x$ and $\boldsymbol{u}_y$, respectively, of all segments, and the corresponding reference components $\bar{\boldsymbol{u}}_x$ and $\bar{\boldsymbol{u}}_y$ for segment 5. The average NRMSE over all spatial nodes is (f) 5.0%, (h) 5.3%.

Controlled Bending Deformations

A tapered robot arm was simulated to track the following time-varying reference configuration, which requires bending deformations:

$$\bar{\boldsymbol{u}}_x(t,s) = \begin{cases} 30\left(\dfrac{s}{L}\right)\sin(\omega t), & k = 1, ..., N/2 \\ 30\left(\dfrac{s-L}{L}\right)\sin(\omega t), & k = N/2+1, ..., N, \end{cases} \tag{3.23}$$

$$\bar{\boldsymbol{u}}_y(t,s) = \begin{cases} 40\left(\dfrac{s}{L}\right)\cos(\omega t), & k = 1, ..., N/2 \\ 40\left(\dfrac{s-L}{L}\right)\cos(\omega t), & k = N/2+1, ..., N, \end{cases} \tag{3.24}$$

and $\bar{\boldsymbol{u}}_z(t,s) = 0$, where $\omega = 2\pi/120$ and $s = k \cdot ds$.

Figures 3.2a-d show snapshots of the simulated hydrogel-based arm over one cycle of the reference input. In these figures and in all subsequent simulation snapshots, only the $N_v = 8$ virtual segments (bounded by black rings on the arm), rather than all $N = 80$

spatial nodes, are displayed for clearer visualization. Each node traces out an ellipse in the $x - y$ plane while exhibiting bending deformations about lines in this plane. Figures 3.2e and 3.2g plot the time evolution of the moments applied by the actuators about the x and y axes. The segments' curvature vector components $\boldsymbol{u}_x$ and $\boldsymbol{u}_y$ are plotted over time in Figs. 3.2f and 3.2h, along with the reference components for segment 5. It is evident that $\boldsymbol{u}_x$ and $\boldsymbol{u}_y$ for this segment remain close to their reference profiles.

Controlled Torsion Deformations

Twisting motions about the z-axis of the robot arm can be simulated by applying a moment about this axis to each spatial node along the arm. We simulated torsion deformation in a tapered robot arm by applying the time-varying reference configuration

$$\bar{\boldsymbol{u}}_z(t, s) = 60 \left(\frac{s}{L} \right) \cos(\omega t), \quad \omega = 2\pi/120, \tag{3.25}$$

and setting $\bar{\boldsymbol{u}}_x(t, s) = \bar{\boldsymbol{u}}_y(t, s) = 0$. Four snapshots of the simulated hydrogel-based arm over one cycle of the reference input are shown in Figs. 3.3a-d. Fig. 3.3e plots the time evolution of the moments applied by the actuators about the z axis in the local frame of each segment. The corresponding curvature vector components $\boldsymbol{u}_z$ are plotted in Fig. 3.3f and compared with the reference component for segment 8.

Controlled Shear Deformations

To our knowledge, shear deformation has not previously been simulated in soft robot arms, despite the development of models such as the one proposed by Renda *et al.* (2016) that are capable of describing shear in such structures. We simulated shear deformation in robot arms with uniform cross-sectional areas. In these simulations, we defined reference shear components $\bar{\boldsymbol{v}}_x$ and $\bar{\boldsymbol{v}}_y$ whose values did not exceed the values of $\boldsymbol{v}_x$ and $\boldsymbol{v}_y$ produced during the bending simulation described in Section 3.1.3, to ensure that they did not exceed

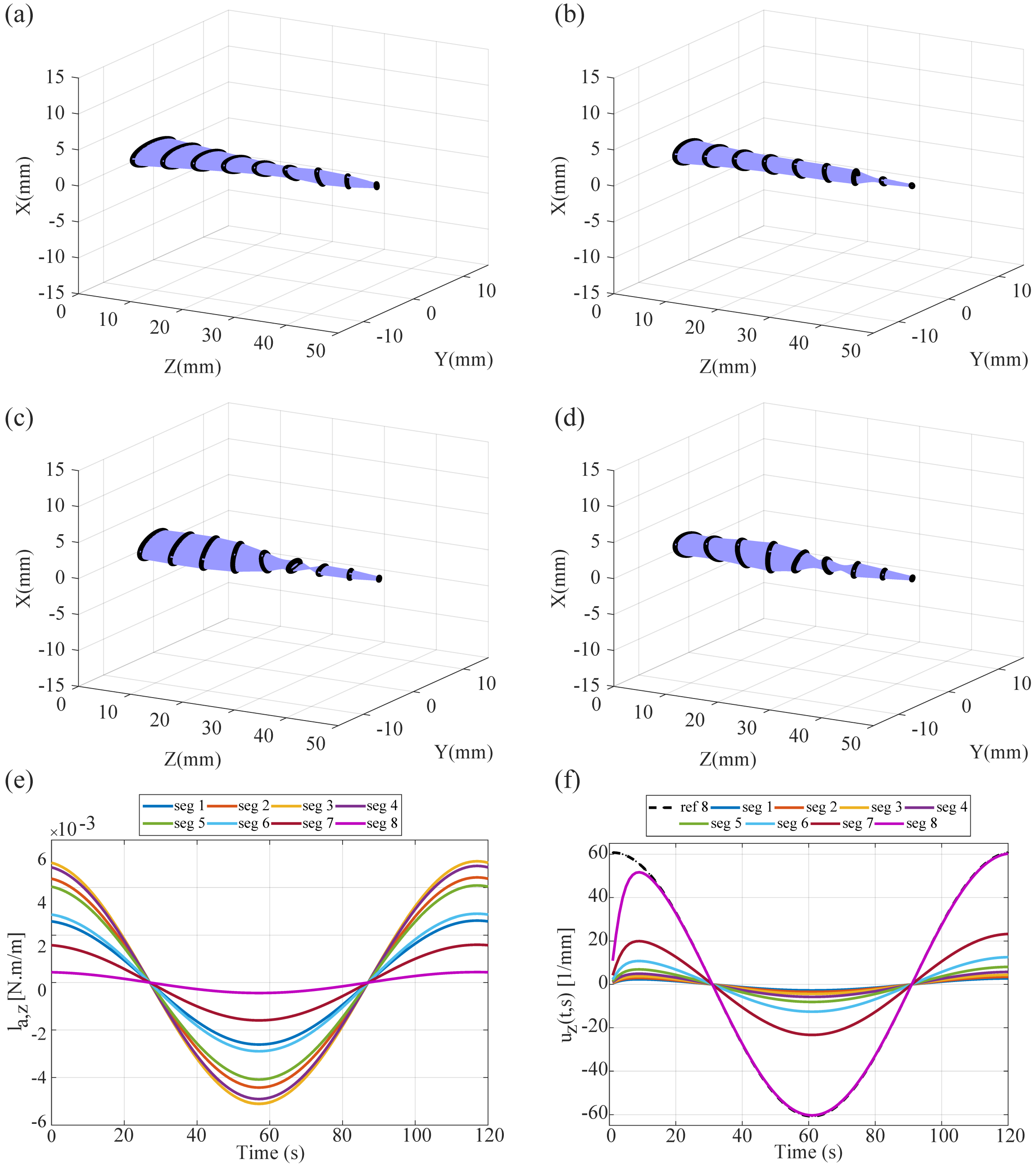


Figure 3.3: Tracking performance for a reference configuration that requires torsion deformation. (a)-(d) Simulation snapshots at $t = 5\,s$, $15\,s$, $30\,s$, and $45\,s$, respectively. (e) Applied moments about the z axis. (f) Curvature components $\boldsymbol{u}_z$ for all segments and $\bar{\boldsymbol{u}}_z$ for segment 8. The average NRMSE over all spatial nodes is 4.54%.

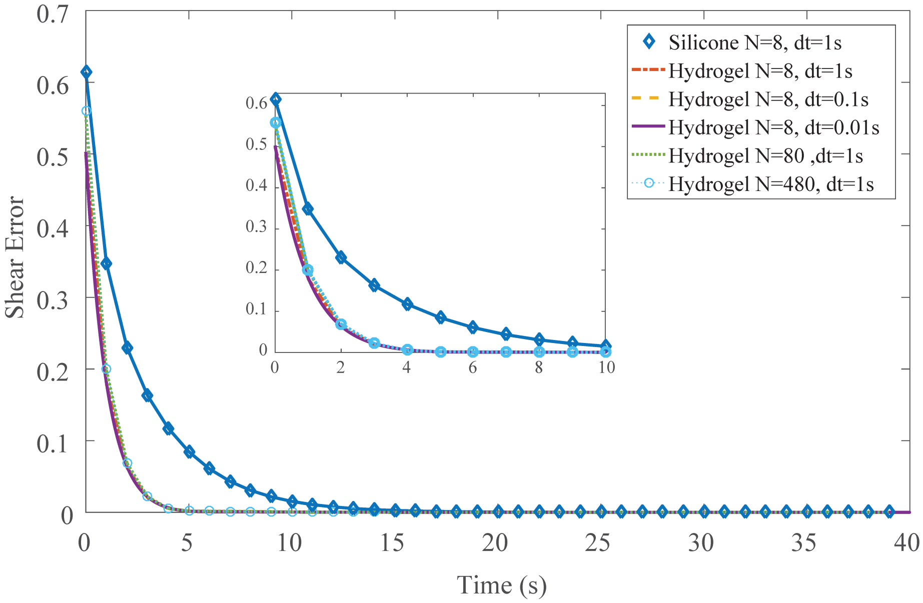


Figure 3.4: Average $||\boldsymbol{e}(t)||_2$ over time of a simulated continuum robot arm undergoing shear deformation, with exponentially decaying reference inputs $\bar{\boldsymbol{v}}_x$ and $\bar{\boldsymbol{v}}_y$ for different values of N, dt, and ρ. The inset figure shows a close-up of the plots during the first $10\,s$.

values that would result in material failure. Our definitions of these reference components are based on the kinematic locomotion of a burrowing worm simulated in one dimension proposed by Boyer *et al.* (2011) and the elongation motions that octopus arms exhibit when their transverse muscles contract discussed by Kolachalama and Lakshmanan (2020); Kier and Smith (1985). The reference components were defined as:

$$\bar{\boldsymbol{v}}_x(t,s) = \bar{\boldsymbol{v}}_y(t,s) = 4\sin\left(\omega\left(t - 1 + \frac{s}{L}\right)\right), \quad \omega = \frac{2\pi c}{\lambda}, \tag{3.26}$$

and $\bar{\boldsymbol{v}}_z(t,s) = 0$. In order to accurately simulate the shear deformations, the cross-sectional area of the arm must be updated at each time step of the simulation with the reference components computed similar to the formula presented by Boyer *et al.* (2011) as follows:

$$A_i = A_0 / \sqrt{\bar{\boldsymbol{v}}_x{}^2 + \bar{\boldsymbol{v}}_y{}^2}, \quad i = 1, ..., N, \tag{3.27}$$

where A_0 is the initial cross-section area.

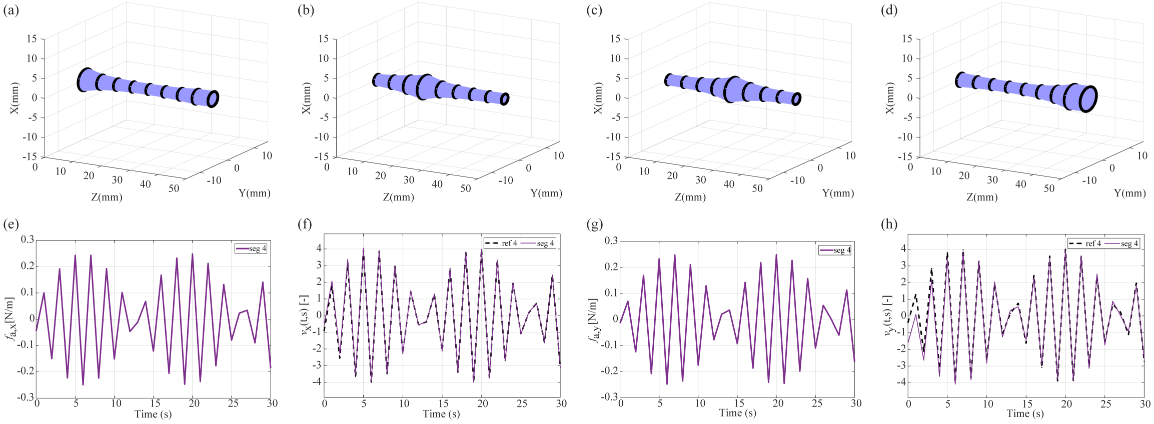


Figure 3.5: Tracking performance for a reference configuration that requires shear deformations. (a)-(d) Simulation snapshots at $t = 1\,s$, $6\,s$, $8\,s$, and $12\,s$, respectively. (e),(g) Applied forces along the x and y axes, respectively. (f),(h) Components $\boldsymbol{v}_x$ and $\boldsymbol{v}_y$, respectively, for segment 4, and the reference components $\bar{\boldsymbol{v}}_x$ and $\bar{\boldsymbol{v}}_y$ for this segment. The average NRMSE over all spatial nodes is (f) 6.15%, (h) 6.3%.

We simulated the responses of hydrogel-based and silicone robot arms to the reference components $\bar{\boldsymbol{v}}_x$ and $\bar{\boldsymbol{v}}_y$. For the hydrogel arm, we set the wave velocity to $c = 1/2\pi$ m/s and the wavelength to $\lambda = L$ m. With these parameters, each spatial node along the robot arm follows a sinusoidal trajectory with equal amplitudes in the x and y directions and a frequency of $\omega \approx 22$ Hz. This frequency was selected in order to visualize the traveling wave response of the robot arm over a short time period ($12\,s$), although in practice, it may be too high to be implemented in hydrogel material. In the silicone arm simulation, the reference input frequency was set to $\omega \approx 2$ Hz. Figures 3.5a-d plot snapshots of the simulated hydrogel arm over one cycle of the reference input. The forces applied by the actuators along the x and y axes are plotted over time in Figs. 3.5e and 3.5g, and the segments' components $\boldsymbol{v}_x$ and $\boldsymbol{v}_y$ and the reference values for segment 4 are plotted over time in Figs. 3.5f and 3.5h.

To study the effects of the number of spatial nodes N, the time step, dt, and the material density, ρ, on the controller performance, shear deformations were simulated in both hydrogel and silicone robot arms for different combinations of N and dt values. For each case, the norm of the shear tracking error, $\boldsymbol{e}(t) = (\bar{\boldsymbol{v}} - \boldsymbol{v}, \mathbf{0})^T$, is plotted over time in Fig. 3.4. Although N and dt must be chosen carefully to avoid producing numerical instability, the results show that the numerical solution of the closed-loop dynamics remains stable over a wide range of N and dt values, without exhibiting a significant variation in performance. Therefore, we selected $dt = 0.1\,s$ and $N = 80$ to achieve acceptable model accuracy without excessive simulation times.

Controlled Extension Deformations

A continuum robot arm can extend and contract along its central axis if its material and design allow the cross-section of the arm to expand and shrink as shown by Guan *et al.* (2020); Morales Bieze *et al.* (2020). These deformations are similar to those produced by muscular hydrostats such as octopus arms presented by Kolachalama and Lakshmanan (2020); Kier and Smith (1985), which elongate or shorten by contracting their transverse or longitudinal muscles while maintaining a constant volume. We simulated extension and contraction in a tapered robot arm by applying the time-varying reference configuration

$$\bar{\boldsymbol{v}}_z(t, s) = 0.75 + 0.25 \cos(\omega t), \tag{3.28}$$

and setting $\bar{\boldsymbol{v}}_x(t, s) = \bar{\boldsymbol{v}}_y(t, s) = 0$. This reference input affects all spatial nodes equally with the same frequency, $\omega = 2\pi/120$ Hz. To enforce a constant volume during extension, mimicking the isovolumetric property of a muscular hydrostat, the cross-sectional area of the arm is updated at each time step of the simulation as $A_i = A_0/\bar{\boldsymbol{v}}_z$, $i = 1, ..., N$. Figures 3.6a-d plot snapshots of the simulated hydrogel-based arm over one cycle of the reference input. Figure 3.6e plots the time evolution of the forces applied by the actuators along the z axis of the local frame of each segment, and Fig. 3.6f compares the segments'

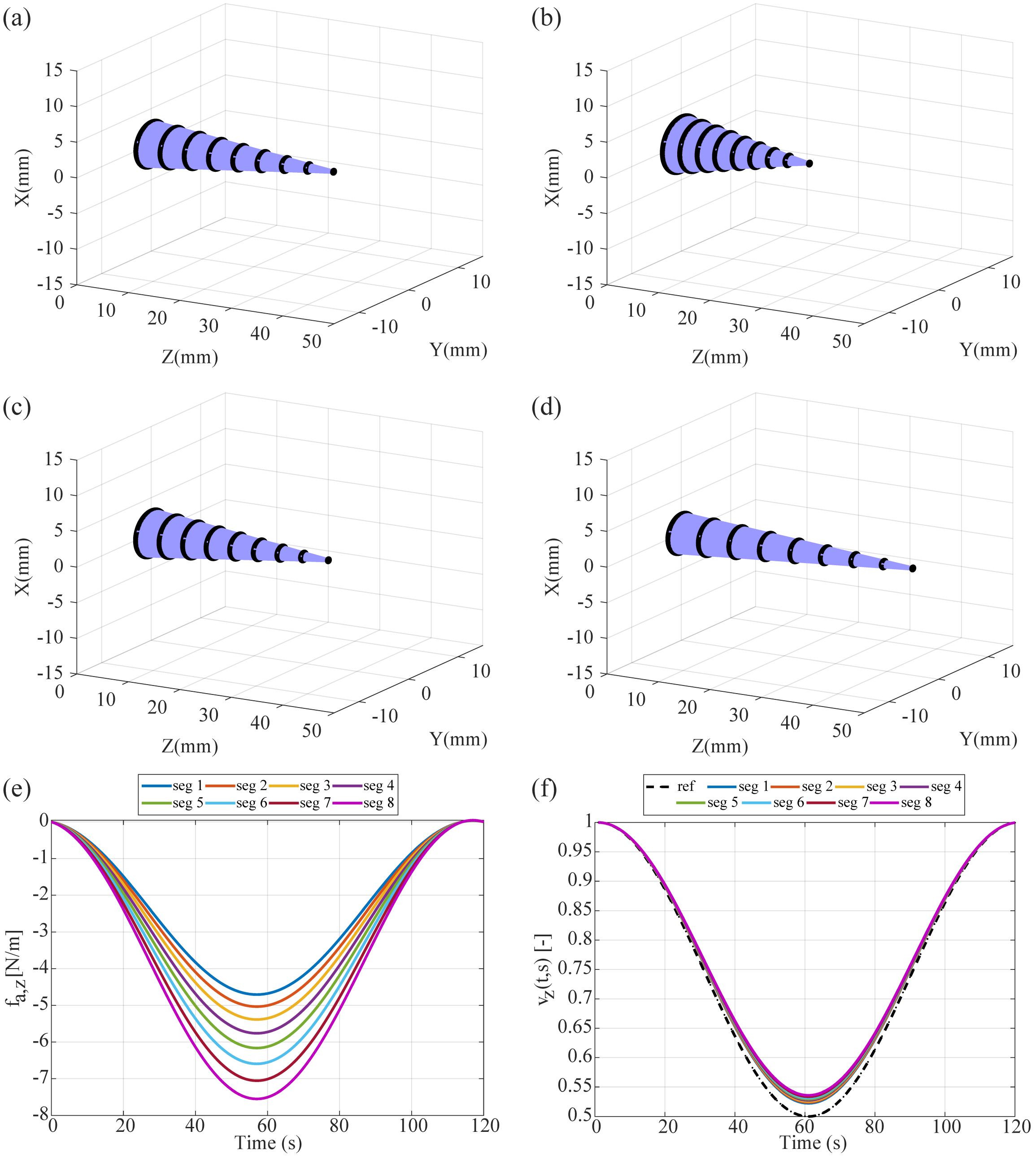


Figure 3.6: Tracking performance for a reference configuration that requires extension deformations. (a)-(d) Simulation snapshots at $t = 30\,s$, $60\,s$, $90\,s$, and $120\,s$, respectively. (e) Applied forces along the z axis. (f) Components $\boldsymbol{v}_z$ and $\bar{\boldsymbol{v}}_z$ for all segments. Average NRMSE over all spatial nodes is 1.3%.

components $\boldsymbol{v}_z$ to the reference input $\bar{\boldsymbol{v}}_z$.

3.1.4 Discussion

We have presented a novel approach to dynamic control of bending, torsion, shear, and extension deformations in a soft continuum robot arm by applying an inverse dynamics controller combined with a decentralized controller that incorporates the system stiffness and damping in the gain matrices. This decentralized control approach can be implemented in a computationally efficient way on a continuum robot arm with independently-controllable actuators enabling it to perform dexterous motions in three dimensions, and it is robust to individual actuator failures. To our knowledge, this is the first control approach for soft continuum robot arms that achieves tracking of configurations which require shear deformations. We validate our controller in simulations of a segmented continuum robot arm that are based on the geometrically exact Cosserat rod model. Our test cases include simulations in which the actuators that apply forces and moments on the elastic rod have the physical properties of a hydrogel material used in Section 3.2. In all simulated test cases, the controller produces average NRMSEs in configuration tracking below 7%, indicating effective tracking performance. The controller's performance is consistent over a wide range of values for the simulation parameters N and dt.

3.2 Development and Validation of a Cosserat Rod Model of a Miniature Hydrogel-based Robotic Arm

3.2.1 Robot Design and Fabrication

As illustrated in Fig. 3.7, the soft continuum robot is a cantilevered arm fabricated from a 3D-printed elastic backbone with orthogonal plates that support 16 cubical soft voxel actuators (SVAs), which are composed of a smart hydrogel material. Each SVA

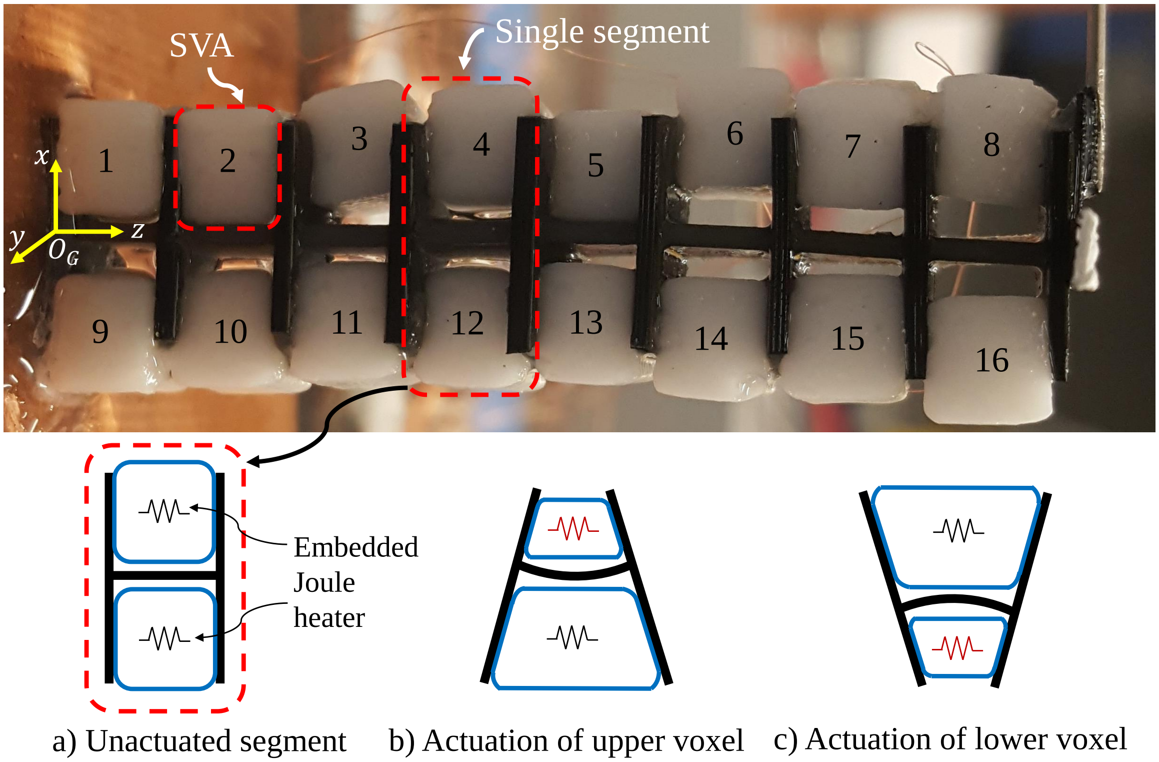


Figure 3.7: *Top:* Soft continuum robot fabricated from a 3D-printed elastic material that supports 16 soft voxel actuators (SVAs) with embedded Joule heaters that actuate the robot. Each segment contains two SVAs. The global coordinate frame is shown in yellow. *Bottom:* Illustrations of a single segment in which: a) both SVAs are in the unactuated state; b) the upper SVA is actuated, causing the backbone to bend upward; c) the lower SVA is actuated, causing the backbone to bend downward.

contracts when it is heated by its embedded Joule heater, and when the heater is turned off, it swells back to its initial shape. The SVA's actuation causes the backbone to bend at the corresponding *segment* (outlined in Fig. 3.7), and as a result, the robot's tip exhibits a deflection along the x-axis. Multiple SVAs can be configured as elastic beams or rods with different numbers of degrees of freedom (DoF). Our ultimate goal is to construct an SVA-acuated robot that can deform in 3D and imitate the four motions of elongation, shortening, bending, and twisting that occur in an octopus arm (Kier, 2016). Our current soft continuum robot has 16 actuators that can produce bending motions in the $x - z$ plane in Fig. 3.7.

In order to fabricate the SVAs, we used standard PNIPAAM hydrogel precursor solution, along with a proprietary recipe described by Khodambashi *et al.* (2019) which was poured into molds using pipettes while the resistive heater (10 Ω SMD resistor 0805) was held in place using grippers. A UV LED (UV 365nm, 10W, Shenzhen Chanzon Technology Co., Ltd., China) was used for curing the gels. After curing, the actuator was placed in a vessel containing deionized (DI) water. The water was changed several times every few hours in order to wash away the solvents. Once fully swollen, each SVA has the dimensions 4.5×4.5×4.5 mm and weighs 0.125 g. To create the hydrogel-based continuum robot, a 4.5×11×45 mm elastic backbone with orthogonal plates was 3D-printed, and the SVAs were affixed to the plates using superglue. When all 16 SVAs are fully swollen, the total mass of the robot is 3 g, and its density (ρ) is $1346.8\,\mathrm{kg/m^3}$.

Each SVA can be selectively commanded as a local actuator. To actuate an SVA, an input voltage is sent to 0.08 mm-diameter copper wires that protrude from the SVA's embedded-Joule heater and are connected to the electrical circuit boards. An Arduino Mega 2560 acts as a communication hub between MATLAB and a MOSFET board (PCA9685 National Control Devices, Osceola, Missouri, U.S.A.). This board has 16 channels, each of which activates one of the 16 actuators of the continuum robot (16 output PWM channels that can produce up to 8 watts of power per channel). The MOSFET board receives the values of the PWM ratio and the channel number and then activates the corresponding Joule heater.

3.2.2 Dynamic Modeling

We use Cosserat rod theory to model the nonlinear dynamics of the hyperelastic hydrogel-based continuum robot. Since this model accounts for the effect of bending, torsion, shear, and extension, it can accurately describe all the motions of the soft continuum robot. Although our current robot can only exhibit bending motions, Cosserat rod theory provides a more accurate dynamic model than the Kirchhoff and Euler-Bernoulli models (O'Reilly, 2017). While the Kirchhoff model is also nonlinear, it only incorporates the effects of shear and extension, and the Euler-Bernoulli model, which is linear, cannot be used to describe large deformations. In our previous work (Doroudchi *et al.*, 2018), we used the Euler-Bernoulli model for decentralized vibration control of a segmented soft continuum robot in simulation, but specified that the robot's length-to-diameter ratio was large enough to satisfy the small-deformation condition.

We use the method presented in Section 3.1 to develop a dynamic model of our continuum robot. Then, similar to the simulated fluidic soft robot considered in Section 5.3 of Till *et al.* (2019), which is subject to distributed forces and moments, our robot experiences distributed forces and moments on its backbone, in part produced by the SVAs when they are actuated individually. The external forces and moments on the backbone are denoted by $\boldsymbol{f}$ and $\boldsymbol{l}$, respectively, and are defined as follows:

$$\begin{aligned} \boldsymbol{f} &= (\rho - \rho_w)A\boldsymbol{g} - \boldsymbol{RCq} \ \odot |\boldsymbol{q}| \ + \ \boldsymbol{f}_{seg} + \boldsymbol{f}_{ext} \ , \\ \boldsymbol{l} &= \boldsymbol{l}_{seg} + \boldsymbol{l}_{ext} \ . \end{aligned} \tag{3.29}$$

The first two terms in the equation for $\boldsymbol{f}$ describe the robot's weight in water and the drag force, respectively. The matrix $\boldsymbol{C} = \boldsymbol{I} * C$ is calculated from the vibration tests discussed in Section 3.2.4. The force and moment applied by each SVA to the backbone of its corresponding segment are denoted by $\boldsymbol{f}_{seg}$ and $\boldsymbol{l}_{seg}$, respectively. The external force and moment on this part of the backbone due to environmental disturbances are denoted by $\boldsymbol{f}_{ext}$ and $\boldsymbol{l}_{ext}$, respectively. We can write $\boldsymbol{f}_{seg}$ and $\boldsymbol{l}_{seg}$, which are expressed in the global frame (O_G), in terms of f_a, the magnitude of an SVA's force in the local frame:

$$\begin{aligned} \boldsymbol{f}_{seg} &= f_a \boldsymbol{R}_s \boldsymbol{e}_3, \\ \boldsymbol{l}_{seg} &= f_a \boldsymbol{R}[(\boldsymbol{v} + \widehat{\boldsymbol{u}}\boldsymbol{r}) \times \boldsymbol{e}_3 + \boldsymbol{r} \times \widehat{\boldsymbol{u}}\boldsymbol{e}_3]. \end{aligned} \tag{3.30}$$

As can be seen from the illustrations in Fig. 3.7, f_a is zero when both SVAs in a segment are unactuated, and it is non-zero when one SVA is actuated. Note that since the robot cannot undergo elongation or shortening in the axial direction, both SVAs of a segment cannot be actuated simultaneously. The dependence of the direction of $\boldsymbol{l}_{seg}$ on the direction of the vector $\boldsymbol{r}$ is determined by the particular SVA that is actuated in the corresponding segment. In order to numerically solve the above set of equations (3.30) for each segment, the value of f_a is required. In the next section, we identify the relationship between the voltage input to an SVA and its applied force on the plate that supports it.

3.2.3 Actuator Identification

The free energy of the hydrogel results from two molecular processes: stretching the hydrogel network and mixing the network with the small hydrogel molecules (Hong *et al.*, 2008). The hydrogel can undergo large deformation in two modes: a fast process with only shape change, and a slow process with both shape and volume changes. Hong *et al.* (2009), define the chemical potential of the solvent and the deformation gradient of the network as the independent variables of the free-energy function, and it is shown that the boundary value problem of the swollen gel is equivalent to that of a hyperelastic solid. To implement their approach, finite-element packages such as ABAQUS or COMSOL are required to analyze the hydrogel dynamics (Wang *et al.*, 2018). These software packages cannot solve the hydrogel dynamic problem in real-time simultaneously with the control program when the robot is actuated, even when using LiveLink$^{\text{TM}}$ for MATLAB.

In order to solve the hydrogel dynamic equations in real-time, we need to either measure the force magnitude f_a in real-time or estimate it from the input voltage. Toward this end, we have designed an experimental setup in order to find the relationship between the input voltage and output force of an SVA. As shown in Fig. 3.8, we measure the tensile force applied by an SVA when it is actuated while bounded by two plates, similar to its configuration on the robot. In this setup, one side of an SVA is glued to a fixed plate at the bottom of a water chamber, and the other side is glued to a force sensor (load cell). The copper wires of the Joule heater are connected to grippers, which are connected to the control board in order to receive an input voltage signal. The data from the force sensor is sent to the computer using a PhidgetBridge 1046_0B bridge interface and a Micro Load Cell (0-100 g) - CZL639HD module.

From the properties of the hydrogel, we know that shape and volume changes occur in the operating region between $25°$C and $32°$C. Considering the resistance of the Joule heater, the range of voltages that we can use to actuate the SVA in order to stay within this operating region is 0 V to 3.7 V. We recorded measurements of the SVA output force over time within this voltage range, at a resolution of 0.18 V, for step input voltages with amplitude V. Since each SVA is affixed to plates that are orthogonal to the backbone of the robot, the SVA applies a tensile force of f_a to these plates when actuated. We applied the voltage for a length of time which ensured that the SVA had fully contracted, and was therefore exerting its maximum possible force.

Figure 3.9 illustrates the measured tensile force produced by the SVA over time when it was actuated with different step input voltage amplitudes for 40 seconds. During the first 40 seconds, the applied voltage heated the SVA, and thereafter the voltage was set to zero and the SVA cooled down as a result of convective heat transfer to the surrounding water, which was kept at a constant temperature of $25°$C. As the figure shows, the SVA force output f_a over time t and the input voltage amplitude V have a proportional relationship: higher forces are produced when higher voltages are applied. We determined an equation for f_a in terms of t and V, for the 20 values of V that we used in our tests, by applying

Table 3.2: Values of Coefficients in Eq. (3.32).

Coefficient	Value	Coefficient	Value	Coefficient	Value
d_{11}	0.0024	d_{21}	0.4900	d_{31}	0.0383
d_{12}	-0.0004	d_{22}	0.1564	d_{32}	0.0002

black-box identification techniques for input-output data (Ljung, 2001b):

$$\begin{aligned} f_a(V,t) = \frac{b_0}{b_2}\Bigg(1 + \mathrm{e}^{-\frac{b_1 t}{2}}\Bigg(\cosh\Big(\sqrt{\frac{{b_1}^2}{4} - b_2}\; t\Big) \\ + \frac{b_1 \sinh\big(\sqrt{\frac{{b_1}^2}{4} - b_2}\; t\big)}{2\sqrt{\frac{{b_1}^2}{4} - b_2}}\Bigg)\Bigg), \quad 0 \le V \le 3.7,\ t \ge 0. \end{aligned} \tag{3.31}$$

We identified the b_i coefficients as the following functions of V using the System Identification and Curve Fitting Toolboxes in MATLAB:

$$\begin{aligned} b_0 &= d_{11}V + d_{12}, \\ b_1 &= d_{21}V^2 + d_{22}, \\ b_2 &= d_{31}V + d_{32}. \end{aligned} \tag{3.32}$$

The values of the identified coefficients d_{ij} are listed in Table 3.2.

Figure 3.10 compares the measured force over time to the identified function for $f_a(V,t)$ in Eq. (3.31) for the case where an SVA is actuated with a 1.08 V step input for 40 seconds. The resulting root-mean-square error (RMSE) between the two plots is 0.0026 N. To compare the RMSE values for the different input voltages, we calculated the normalized root-mean-square error (NRMSE) for each voltage. This error was relatively low, under 5%, for all input voltage amplitudes that were tested. We leave model validation for more complicated input voltages for future work on closed-loop control of the robot.

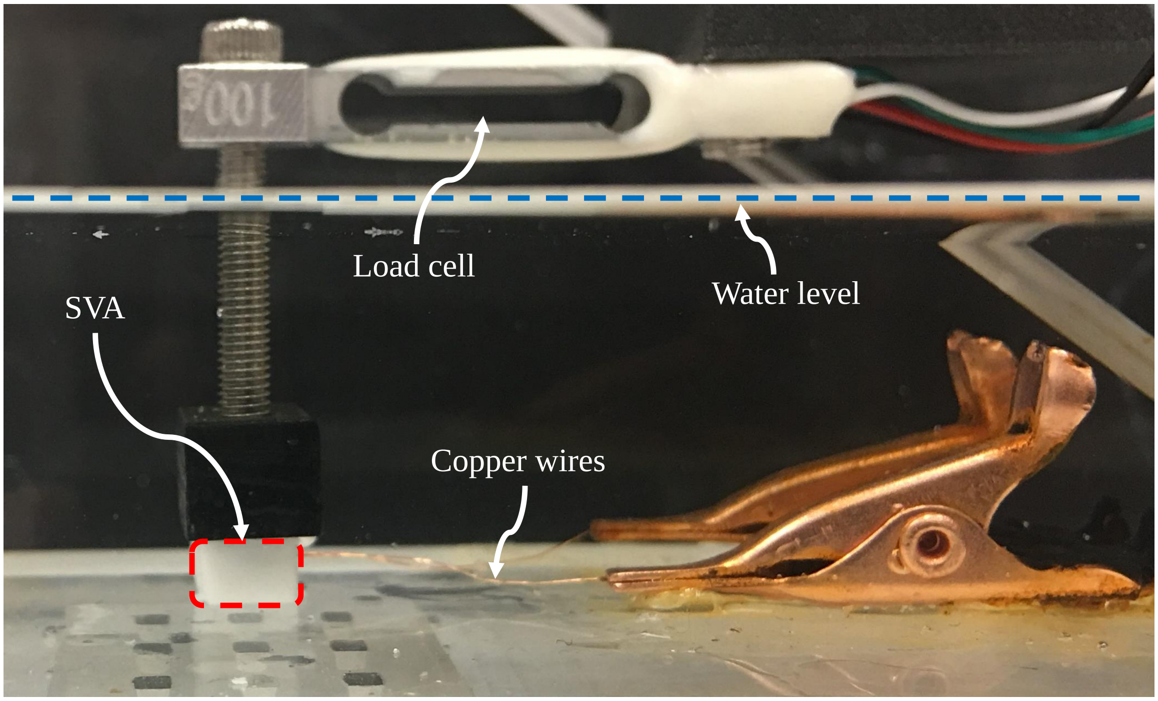


Figure 3.8: Experimental setup that uses a load cell to measure the force applied by an SVA.

3.2.4 Experimental Validation and Discussion

In this section, we validate the Cosserat dynamical model from Section 3.2.2 by comparing simulations of the model to experimental data for open-loop actuation of the robot. Some parameters of the model are estimated experimentally using vibration tests.

3.2.5 Vibration Tests

The Young's modulus and physical damping coefficient of the robot are not readily available, since the robot is in reality neither homogeneous nor isotropic. Moreover, the submergence of the robot in water during the experiments affects the damping coefficient. In order to estimate the unknown parameters in the dynamical model, we make the approximation that the robot is homogeneous and isotropic. We perform the widely-used weight

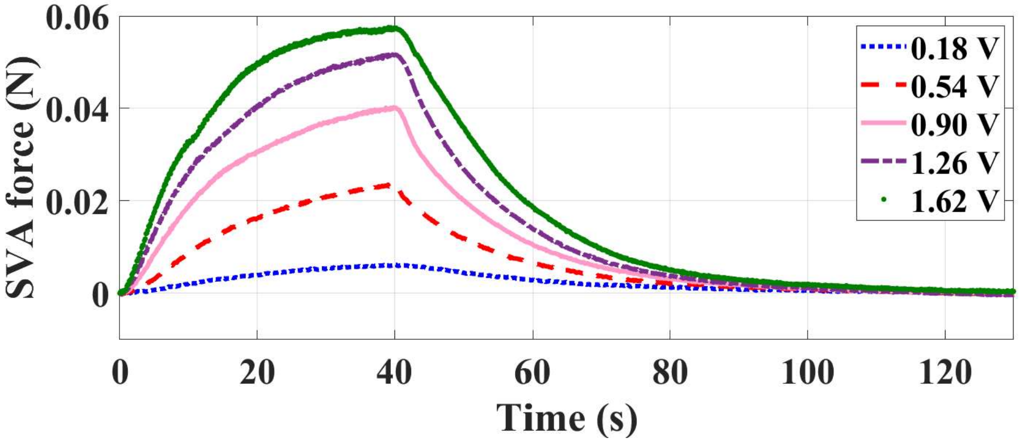


Figure 3.9: Measured force applied by an SVA while it is actuated by different step input voltages for 40 s.

(or force) experiment (Linn *et al.*, 2013) with different values of weights to produce vibrations of the robot's tip until it reaches its equilibrium position. The vibration tests are performed on the robot with the SVAs in their fully swollen state. Figure 3.11 shows side and front views of the robot bending when a 0.405 g load is placed on its tip during a vibration test. A Logitech C930e webcam placed in front of the chamber is used to track the white square marker on the robot's tip, labeled in Fig. 3.11, in order to measure the tip's position over time.

Figure 3.12 plots the robot's vibrations around equilibrium in five cases. In case 1, it is deflected upward and then released to vibrate freely, and in cases 2 through 5, a load of 0.122 g, 0.222 g, 0.325 g, and 0.405 g, respectively, is rapidly added to its tip. To clearly compare the time responses of the robot's deflection, its equilibrium position along the x-axis (see Fig. 3.7) for each case is shifted to 0 mm. The actual equilibrium positions for cases 1 to 5 were -0.29, -4.25, -4.71, -7.39, and -7.89 mm, respectively.

We measured the amplitudes of the vibrations at adjacent peaks for different cases, and determined the damping ratio from the logarithmic decrements of these measure-

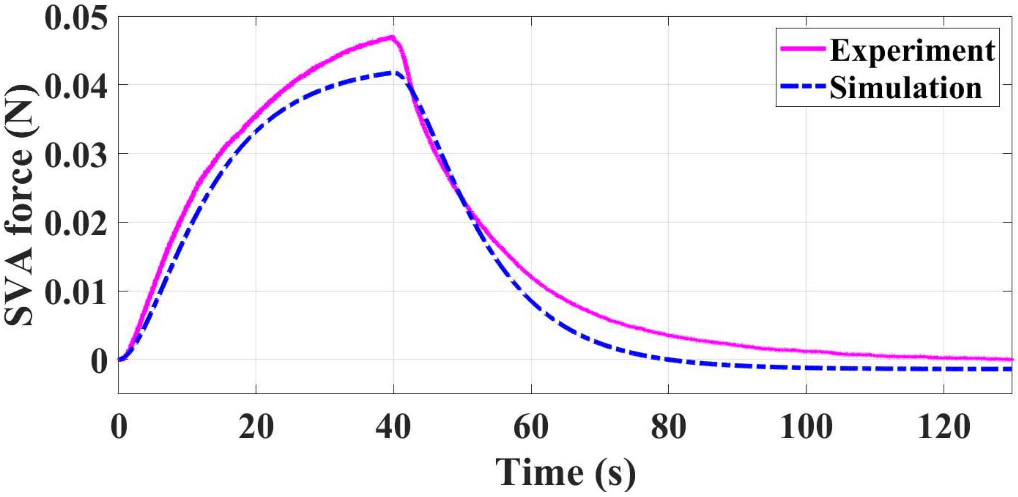


Figure 3.10: Simulation of the identified force function (3.31) compared to the experimentally measured force for the case of an SVA actuated by a 1.08 V step input for 40 s.

ments (Lang *et al.*, 2011). Using the structural damping equations for cantilevered beams and spring-mass systems, the damping coefficient and Young's modulus values were calculated as $0.0262\,\mathrm{kg/m^2}$ and 2.005 kPa, respectively.

3.2.6 Verification of the Dynamical Model

Having estimated the unknown parameters and measured the SVA applied tensile forces, we were able to simulate the robot forward dynamics in MATLAB using Eqs. (3.1)–(3.2)–(3.32). Every segment's position and orientation were calculated at each iteration of the simulation. We tested eight cases in both simulation and experiment, in which the robot was actuated in an open-loop manner with square signals of voltage that were applied to different subsets of SVAs. Table 3.3 lists the parameters for our eight test cases, including the numbers of the particular SVAs that are actuated with their corresponding input voltages, which were all applied for 15 seconds. The RMSE values in the table give the errors between the robot's tip position in experiment and simulation during one test for each case,

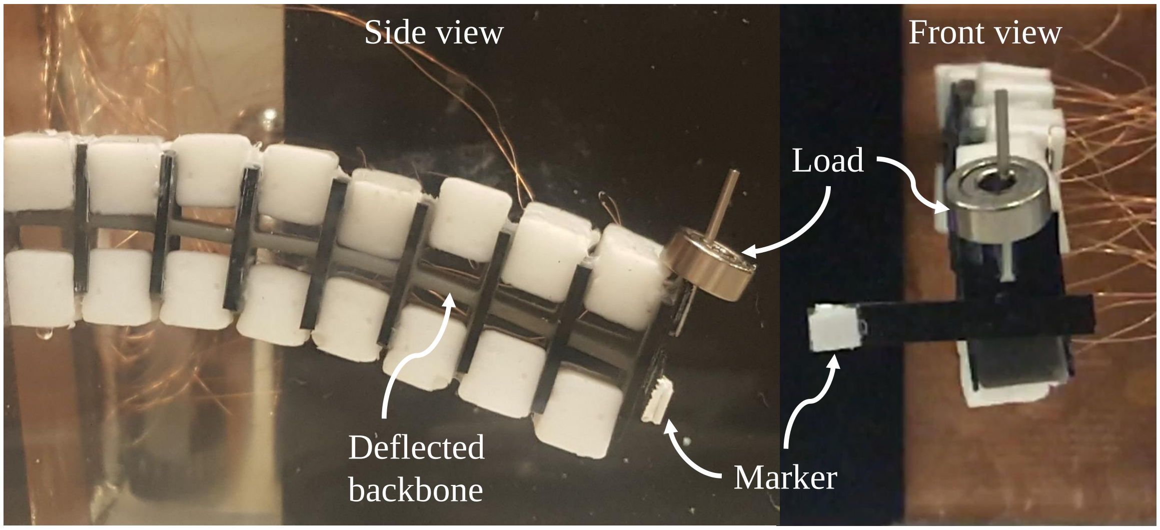


Figure 3.11: Side and front views of the continuum robot during a vibration test in which a 0.405 g load is placed on its tip.

and $\overline{X} = X_{max} - X_{min}$ indicates the tip's maximum displacement from its initial position. Dividing RMSE by $|\overline{X}|$ yields the NRMSE value, for which the percentages are reported. These relatively low NRMSE values, which are all under 10%, indicate the accuracy of the modeling approach. Figures 3.13(a) and 3.13(b) show the simulated and experimental deformations of the robot for Case 4, where SVAs number 9 to 16 were actuated by 3.6 V for 15 s.

Figure 3.14(a) plots the simulated and experimentally measured displacement of the robot's tip for the case when all SVAs in the top row (numbered 1 to 8) are actuated with a square wave of amplitude 1.8 V for 60 seconds and then 0 V for 60 seconds. The RMSE between the plots is 0.68 mm. Figure 3.14(b) plots the simulated and measured displacements for the case when SVA numbers 4, 9, and 15 are actuated with a square wave of amplitude 3.6 V for 15 seconds. The RMSE between the plots for this case is 0.32 mm. Since the delay in the hydrogel response is not included in the SVA force model or the robot dynamical model, at the beginning of the SVA heating state or cooling state and in

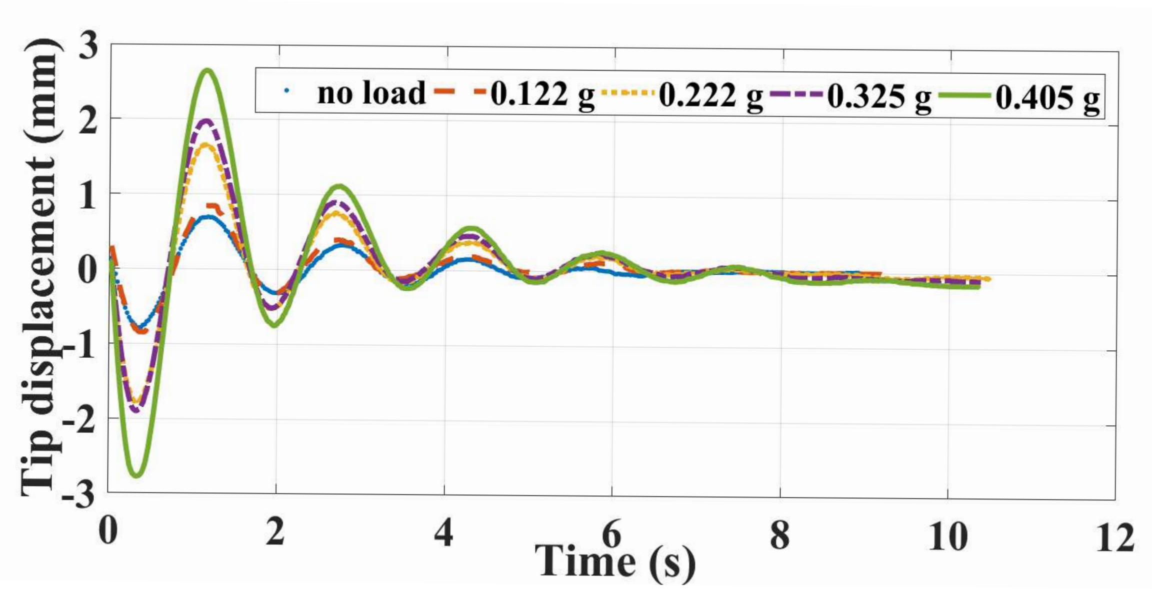
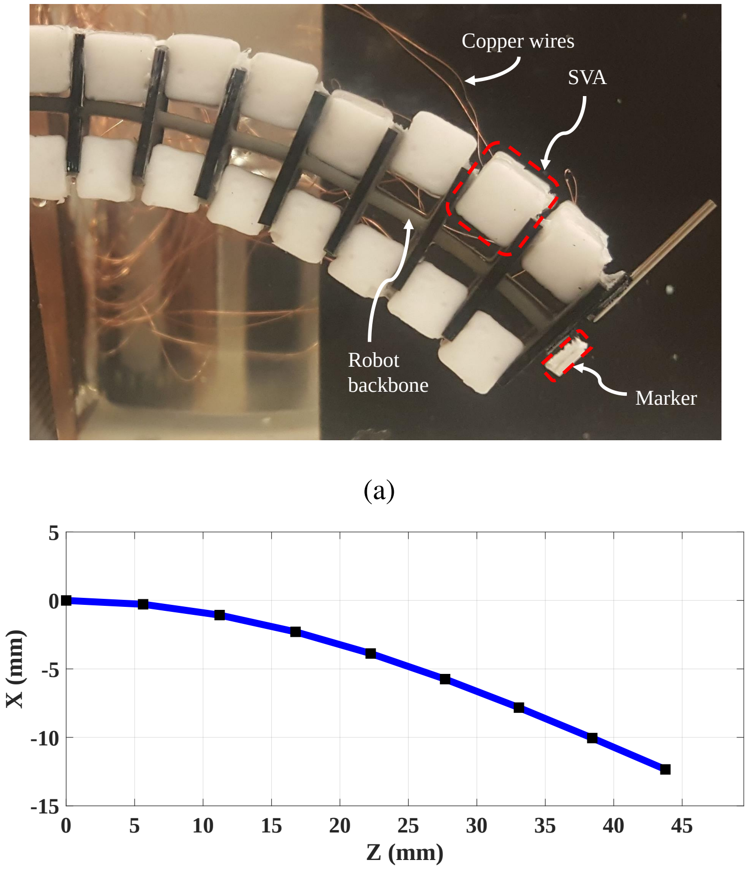


Figure 3.12: Vibrations of the robot's tip around its equilibrium position (shifted to 0 mm) for cases with no weight attached and with attached weights of 0.122 g, 0.222 g, 0.325 g, and 0.405 g. The period of the vibrations is 1.56 s.

general during any change of states, error arises between the simulations and experimental measurements, as is apparent from the results in Fig. 3.14. Other sources of error between the simulations and experiments include deviations from the assumption of a homogeneous and isotropic robot, inconsistencies in SVA fabrication (which can be addressed in future work), and variable hydrogel stiffness during the contraction and swelling modes.

3.2.7 Discussion

In this section, a new hydrogel-actuated continuum robotic arm is presented, and its dynamics are modeled based on the Cosserat theory of rods. The robot is designed to bend when one or more of the actuators located along its length, called SVAs, are actuated by embedded Joule heaters. To measure the force applied by the SVAs during actuation, an experimental setup was designed to obtain input-output data on the voltage-to-force rela-



(a)

(b)

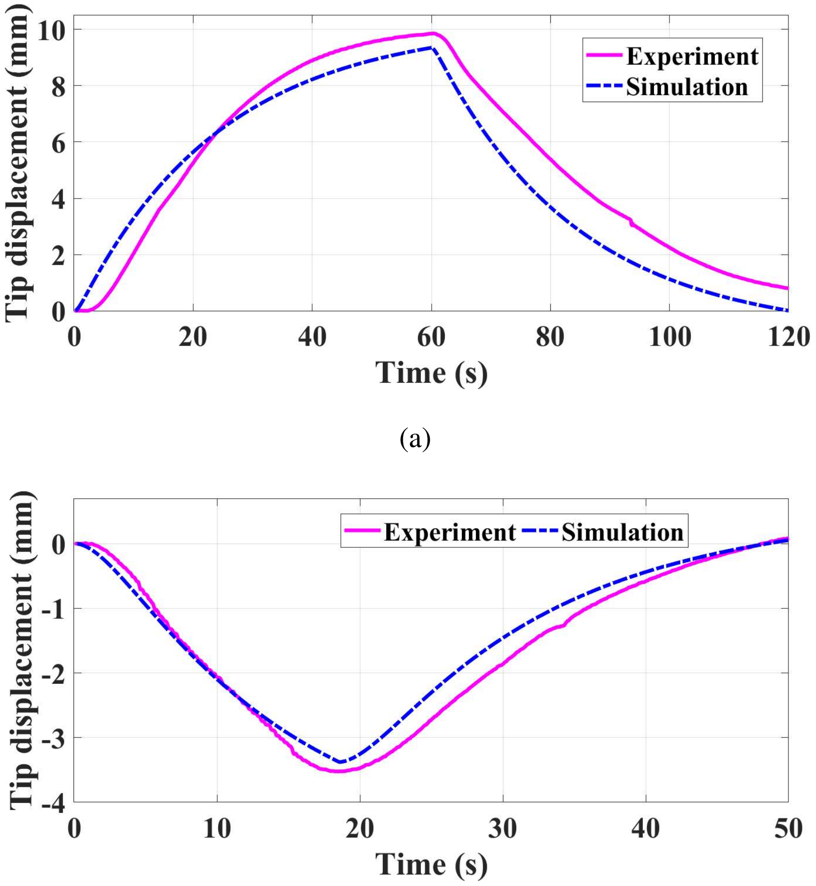
Figure 3.13: Continuum robot bending when its bottom row of SVAs (numbers 9 to 16) have been actuated by a square wave with amplitude 3.6 V applied for 15 seconds (Case 4, Table 3.3): (a) experiment; (b) simulation.

tionship, which was characterized using black-box identification techniques. Experimental validation results demonstrate that the identified model has an NRMSE of at most 5% over a range of step input voltages (0 V–3.7 V). The dynamic model was also validated in MATLAB simulations after estimating unknown mechanical parameters of the robot in an experimental vibration test. By comparing the physical robot's deformation under open-loop control with predefined actuation patterns to the corresponding simulation results, we show that the model accurately reproduces the robot dynamics, with at most 10% NRMSE. In fu-

Table 3.3: Errors Between Measured and Simulated Robot Tip Displacements.

Case number	Actuators number	Voltage (V)	RMSE (mm)	$\lvert\overline{X}\rvert$ (mm)	NRMSE (%)
1	1-8	1.8	0.28	4.09	6.8%
2	1-8	3.6	0.69	8.96	7.7%
3	9-16	1.8	0.39	6.51	5.9%
4	9-16	3.6	0.63	13.06	4.8%
5	3,4,5	3.6	0.36	4.98	7.2%
6	14,15,16	3.6	0.16	2.01	7.9%
7	7,8,11,12,13	3.6	0.49	5.15	9.5%
8	4,9,15	3.6	0.32	3.52	9.1%

ture work, we will incorporate shear and torsion effects into our models for new designs of hydrogel-based continuum robots that can deform in 3D. Moreover, we will develop decentralized controllers for such robots, with the goal of performing more complex tasks such as position control of the tip, trajectory tracking, shape control, and object manipulation.



(a)

(b)

Figure 3.14: Simulated and measured robot tip displacement along the x-axis when (a) SVA numbers 1 to 8 are actuated with a square wave of amplitude 1.8 V applied for 60 seconds; (b) SVA numbers 4, 9, and 15 are actuated with a square wave of amplitude 3.6 V for 15 seconds (Case 8, Table 3.3).

3.3 Decentralized Tracking Control of a Silicone Multi-Segment Robotic Arm using a Cosserat Rod Model

Controlling soft continuum robotic arms is challenging due to their hyper-redundancy and dexterity. Controllers for such robots that are based on dynamical models are generally known to be more accurate compared to other methods; however, their high computational and sensing requirements have limited their usage (George Thuruthel *et al.*, 2018; Webster III and Jones, 2010b). In this section, we demonstrate, for the first time, closed-loop control of the configuration of a silicone multi-segment robotic arm using a Cosserat rod model of the robot. Our controller solves the inverse dynamic problem by simulating the Cosserat rod model in MATLAB using a computationally efficient numerical solution scheme, and applies the computed control output to the actual robot in real-time. The controller is implemented on a two-segment silicone robotic arm with distributed pneumatic actuation, using external vision-based feedback on the segments' configurations. The controller is designed to track a desired reference configuration by independently controlling each segment of the robotic arm using a decentralized approach. A motion capture system is used to measure the position and orientation of the tip of each segment in real-time, and the remaining unknown parameters that are needed to solve the inverse dynamics are estimated simultaneously in the simulation. Thus, this method reduces the complexity of the dynamic controller by reducing the number of required sensors on the robot. The controller is used to reshape the robotic arm into configurations that are achieved through different combinations of bending and extension deformations in 3D space. In all test cases, the controller produces normalized root-mean-square errors (NRMSEs) in configuration tracking that remain below 9%, which shows the effectiveness of the controller and the accuracy of the simulated model that is used to estimate the unmeasured parameters.

3.3.1 Silicone Segment Fabrication and Properties

The silicone multi-segment soft arm was fabricated using a process similar to the one described by Nguyen *et al.* (2017). Each segment consists of four independent cylindrical fiber-reinforced actuators (FRAs) that are arranged in parallel and are symmetric with respect to the central axis of the segment. Both clockwise and counterclockwise helically winded threads allow axial elongation of the FRAs while preventing radial expansion. Once

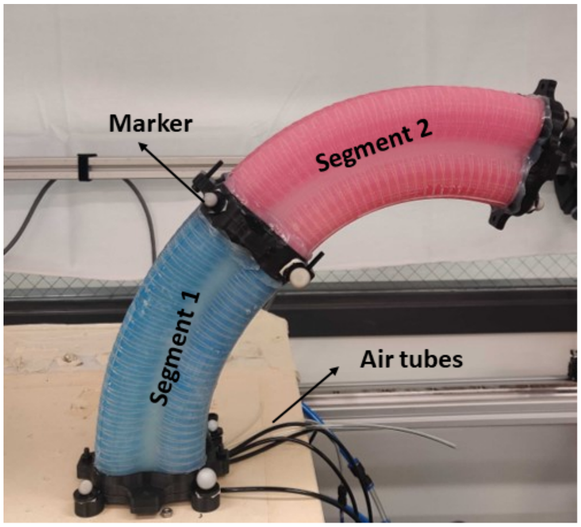


Figure 3.15: The multi-segment silicone soft robotic arm. The silicone segments were fabricated using the method described in Nguyen *et al.* (2017).

the FRAs are fabricated, a different mold is used to embed all four actuators in a segment. Both silicone material and dye (blue or red) are poured into the mold to fuse the FRAs into a blue or red segment, as depicted in Fig. 3.15. The segments are connected with black 3D-printed parts, which are fastened by bolts and nuts. The source of actuation is compressed air, which is applied to each segment through four air tubes that are each connected to an FRA at one end and to a digital pressure regulator at the other end. The digital pressure regulator measures the air pressure p_m in each FRA and drives it to a desired pressure set-point p_d.

The physical properties of each segment are listed in Table 3.4. In order to estimate the Young's modulus of a segment, assuming that its value remains constant during actuation, an Instron 5944 universal testing machine was used to exert a force on an unactuated segment under three loading speeds (1 mm/s, 3 mm/s and 5 mm/s) and measure its corresponding change in length. Three trials were conducted for each loading speed, and the Young's modulus was estimated as the average value over all 9 trials.

Table 3.4: Silicone Segment Properties.

Param.	Description	Value	Units
N	Number of segments	2	–
r_i	Distance from backbone to each FRA	0.3	cm
A_i	FRA cap area	3.1	cm^2
r_0	Undeformed radius of the segment	5.3	cm
L_0	Undeformed length of the segment	18.5	cm
M	Mass of the segment	0.825	kg
ρ	Density of the segment	792.8	kg/m^3
G	Shear modulus of the segment	0.1	MPa
E	Young's modulus of the segment	0.28	MPa

3.3.2 Forward Dynamics of the Multi-Segment Arm based on the Cosserat Rod Model

As mentioned in Section 3.1.1, the equations of an elastic Cosserat rod are written for its central backbone, and any forces and moments that are applied to this backbone are modeled as external forces and moments. In practice, the sources of these external forces and moments (${}^G\boldsymbol{f}, {}^G\boldsymbol{l}$) are the control inputs applied by the pneumatic actuators of the continuum robot arm (${}^G\boldsymbol{f}_p, {}^G\boldsymbol{l}_p$) and environmental effects (${}^G\boldsymbol{f}_e, {}^G\boldsymbol{l}_e$) such as gravitational and drag forces:

$$\begin{aligned} {}^G\boldsymbol{f} &= {}^G\boldsymbol{f}_p + {}^G\boldsymbol{f}_e, \\ {}^G\boldsymbol{l} &= {}^G\boldsymbol{l}_p + {}^G\boldsymbol{l}_e. \end{aligned} \tag{3.33}$$

The only environmental effect here is the gravitational force per unit length:

$${}^G\boldsymbol{f}_e = \rho A {}^G\boldsymbol{g}, \tag{3.34}$$

where ${}^G\boldsymbol{g} = [0\ 0\ -9.81]^T\, m/s^2$. Thus, the pneumatic actuators' applied forces and

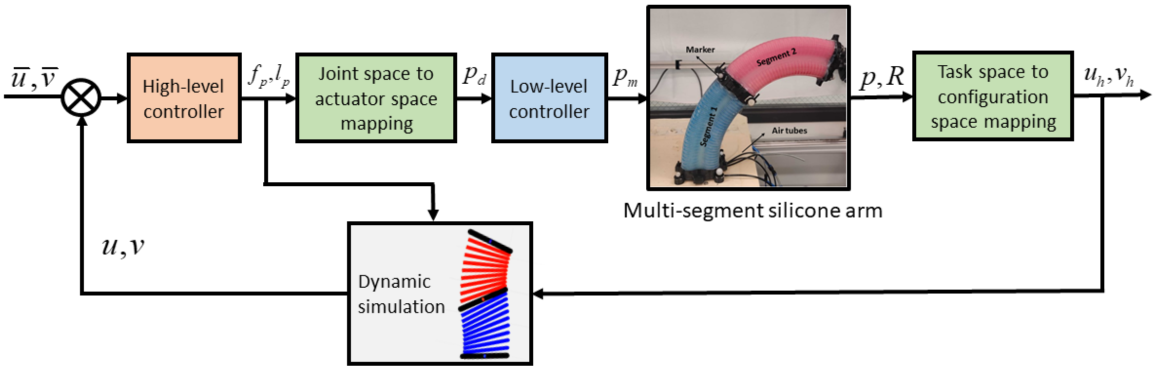


Figure 3.16: Schematic block diagram of the controller.

moments to the backbone are defined as:

$$\begin{aligned} {}^G\boldsymbol{f}_p &= \sum_{i=1}^{4} p_i A_i [{}^G\boldsymbol{R}\boldsymbol{e}_3] - \rho A {}^G\boldsymbol{g}, \\ {}^G\boldsymbol{l}_p &= \sum_{i=1}^{4} p_i A_i [({}^G\boldsymbol{p} + {}^G\boldsymbol{R}\boldsymbol{r}_i) \times {}^G\boldsymbol{R}\boldsymbol{e}_3], \end{aligned} \tag{3.35}$$

where p_i is the chamber air pressure of the i-th FRA, A_i is the corresponding chamber cap area, and $\boldsymbol{r}_i$ is the vector from the backbone to the center of the i-th FRA in the local frame of a segment cross-section (Figure. 3.17). The gravitational effect is subtracted from the actuation effect to cancel out its effect on the backbone.

By substituting (3.33) into (3.1), the relationship between the applied forces and moments from the pneumatic actuators and the virtual joint space variables ($\boldsymbol{n}_s, \boldsymbol{m}_s$) at each discretized spatial location (node) is defined. Then, the configuration state variables $\boldsymbol{v}$ and $\boldsymbol{u}$ are computed from the joint space variables, $\boldsymbol{n}$ and $\boldsymbol{m}$, using (3.8). The history elements of the configuration variables, $\boldsymbol{v}_h$ and $\boldsymbol{u}_h$, are obtained from experimental data on the position and orientation of the robot in its task space, as discussed in the following section.

3.3.3 Decentralized Approach to Inverse Dynamic Control of the Multi-Segment Arm

Figure 3.16 illustrates a schematic block diagram of our proposed controller for configuration tracking control of the soft multi-segment robot arm. The time-varying reference configuration variables for bending and extension deformations are sent to the control loop at each time step. The high-level controller computes the control output based on the difference between the current configuration variables of the robot and the desired values. The control outputs, which are the desired forces and moments applied by the actuators, are mapped to the desired air pressure p_i in each FRA. Then, the desired pressures are sent to the robot's pneumatic actuators using a low-level controller. The position and orientation of the tip of each segment is measured by an Optitrack motion capture system with Motive software.

To solve the inverse dynamics problem, the silicone segment's bending and extension deformations in the task space are mapped to the configuration state variables $\boldsymbol{v}_z$,$\boldsymbol{u}_x$, and $\boldsymbol{u}_y$. Since the task space to configuration space mapping is robot-independent (Webster III and Jones, 2010b), the Piecewise Constant Curvature (PCC) configuration space variables, (κ, ϕ, L), are used to complete the mapping. Figure 3.17 illustrates the kinematics of a constant-curvature segment that approximates the silicone segment under consideration. The position (x, y, z) of the tip of each segment with respect to the local frame attached to the base of the segment is related to the corresponding PCC configuration variables as follows, defined as in (Jiang *et al.*, 2020):

$$\kappa = \begin{cases} \dfrac{2x}{x^2+z^2}, & \phi = 0 \\ \dfrac{2y}{y^2+z^2}, & \phi = \pi/2 \\ 0, & x = 0,\ y = 0 \end{cases} \tag{3.36}$$

$$L = \begin{cases} \dfrac{1}{\kappa}\arctan\left(\dfrac{x}{z}\right), & \phi = 0 \\ \dfrac{1}{\kappa}\arctan\left(\dfrac{y}{z}\right), & \phi = \pi/2 \\ z, & x = 0,\ y = 0 \end{cases} \tag{3.37}$$

Now, a mapping is required between the PCC configuration space variables defined for the robotic arm and the corresponding configuration state variables in the Cosserat model, since the $(\boldsymbol{v}, \boldsymbol{u})$ variables cannot be directly measured from the robot's task space. The curvature variable in each fixed plane in the PCC model is equal to the component of the curvature vector $\boldsymbol{u}$ in the same plane for bending deformations, and the extension variable v_z is given by the extension ratio of the segment:

$$\begin{aligned} u_y &= \kappa \quad for \quad \phi = 0, \\ u_x &= \kappa \quad for \quad \phi = \pi/2, \\ v_z &= 1 + \frac{L - L_0}{L_0}, \end{aligned} \tag{3.38}$$

where L_0 is the undeformed length of the segment.

Using the mapping from (3.38), the configuration state variables are computed from the measured task space variables. Lastly, the values of the configuration state variables at the next time step are approximated by the simulation of the Cosserat model forward dynamics, which are numerically solved given the position and orientation measurements and the high-level controller output. Thus, the inverse dynamics of the robot are solved in the simulation using the actual robot's position and orientation measurements and approximations of the other variables from the Cosserat model forward dynamics solution. The controller is defined as the controller presented in Section 3.1. It is designed with a decentralized structure that has distinct proportional-derivative (PD) gains at each location along the arc length of the Cosserat rod. The controller tracks a static or time-varying reference configuration, $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}})$, that results in bending, extension, or a combination of these deformations in

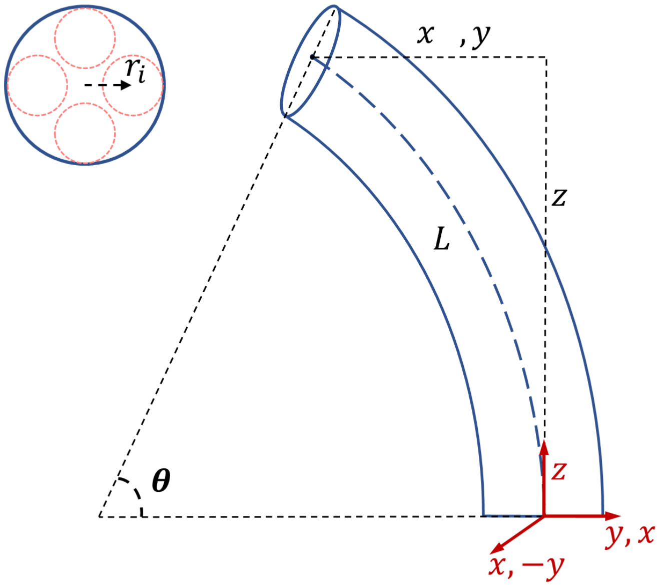


Figure 3.17: Kinematic representation of a constant-curvature segment. $\boldsymbol{r}_i$ is the vector from the backbone to the center of the i-th FRA chamber, which has cap area A_i, in the local frame of a segment cross-section, which has area A.

3D space, and compensates for any environmental effects.

The control outputs are defined as:

$$
\begin{aligned}
{}^G\boldsymbol{f}_p &= {}^G\boldsymbol{R}[\boldsymbol{K}_{v_1}(\bar{\boldsymbol{v}}_t - \boldsymbol{v}_t) + \boldsymbol{K}_{p_1}(\bar{\boldsymbol{v}} - \boldsymbol{v}) + \boldsymbol{K}_{m_1}\bar{\boldsymbol{v}}_{tt}] - \rho A\,{}^G\boldsymbol{g}, \\
{}^G\boldsymbol{l}_p &= {}^G\boldsymbol{R}[\boldsymbol{K}_{v_2}(\bar{\boldsymbol{u}}_t - \boldsymbol{u}_t) + \boldsymbol{K}_{p_2}(\bar{\boldsymbol{u}} - \boldsymbol{u}) + \boldsymbol{K}_{m_2}\bar{\boldsymbol{u}}_{tt}],
\end{aligned}
\tag{3.39}
$$

where $\boldsymbol{K}_{m_1}, \boldsymbol{K}_{m_2}$ are 3×3 diagonal matrices whose diagonal entries are proportional to $\rho A, \rho\boldsymbol{J}$, respectively, and $\boldsymbol{K}_{v_1}, \boldsymbol{K}_{v_2}, \boldsymbol{K}_{p_1}, \boldsymbol{K}_{p_2}$ are gain matrices defined as in (3.11) with the scalar coefficients set to $v_1 = v_2 = 1$, $p_1 = 20$, and $p_2 = 25$. The stiffness and damping variables $(K_{se}, K_{bt}, B_{se}, B_{bt})$ are computed from the silicone segment properties in Table 3.4. By substituting the control output from (3.39) into (3.35), the equivalent actuation for each segment, $\boldsymbol{P} = [P_x, P_y, P_z]$, can be computed using the mapping below, given that the silicone segments have only three DOFs: bending about the x-axis, bending about the

y-axis, and elongation along the z-axis:

$$
\begin{aligned}
P_x &= {}^G\boldsymbol{l}_p(y) \cdot {}^G\boldsymbol{l}_p^T(y)/A_i,\\
P_y &= {}^G\boldsymbol{l}_p(x) \cdot {}^G\boldsymbol{l}_p^T(x)/A_i,\\
P_z &= {}^G\boldsymbol{f}_p(z) \cdot {}^G\boldsymbol{f}_p^T(z)/A_i,
\end{aligned}
\tag{3.40}
$$

where A_i is equal for all the chambers. Then, the equivalent actuation is obtained from the real actuator pressures, p_1, p_2, p_3, p_4, as follows:

$$
\begin{aligned}
P_x &= -p_1 + p_2 - p_3 + p_4,\\
P_y &= p_1 + p_2 - p_3 - p_4,\\
P_z &= p_1 + p_2 + p_3 + p_4.
\end{aligned}
\tag{3.41}
$$

Since our silicone segment is similar in design to the HPN Arm (Jiang *et al.*, 2021), in which each segment has two bending DOFs and one elongation DOF, the relation $p_1 + p_4 = p_2 + p_3$ must hold. Therefore, the real actuator pressures are computed as:

$$
\begin{aligned}
p_1 &= (-P_x + P_y + P_z)/4,\\
p_2 &= (P_x + P_y + P_z)/4,\\
p_3 &= (-P_x - P_y + P_z)/4,\\
p_4 &= (P_x - P_y + P_z)/4.
\end{aligned}
\tag{3.42}
$$

Algorithm 5 briefly describes our implementation of this configuration tracking controller on the silicone multi-segment robotic arm. First, the desired time-varying configuration (a combination of bending and extension deformations) is defined in terms of the variables $\bar{\boldsymbol{u}}^{\boldsymbol{i}}_{\boldsymbol{x}}$, $\bar{\boldsymbol{u}}^{\boldsymbol{i}}_{\boldsymbol{y}}$, and $\bar{\boldsymbol{v}}^{\boldsymbol{i}}_{\boldsymbol{z}}$ (line 1). We define an *outer loop* that iterates over time steps (lines 2 to 11) and an *inner loop* that iterates over discretized spatial locations (nodes) along the backbone of the robot (lines 4 to 10). In the outer loop, the initial boundary condition values of $\boldsymbol{n}_0^i$ and $\boldsymbol{m}_0^i$ are guessed using SSM; in the first iteration, they are set to zero (line 3). By applying RK4 to ($\boldsymbol{n}_{j-1}^i$,$\boldsymbol{m}_{j-1}^i$) and the derivatives of the internal force and moment

Algorithm 5 Configuration tracking controller on the silicone multi-segment robotic arm

```
 1: Given a desired configuration $\bar{\boldsymbol{u}}_{\boldsymbol{x}}^{\boldsymbol{i}}, \bar{\boldsymbol{u}}_{\boldsymbol{y}}^{\boldsymbol{i}}, \bar{\boldsymbol{v}}_{\boldsymbol{z}}^{\boldsymbol{i}}$
 2: for $i \leftarrow 1$ to $T/dt$ do
 3:     $\boldsymbol{n}_0^i, \boldsymbol{m}_0^i \leftarrow$ SSM based on (3.7) ($\boldsymbol{n}_L^{i-1} = \boldsymbol{m}_L^{i-1} = \boldsymbol{0}$)
 4:     for $j \leftarrow 1$ to $L/ds$ do
 5:         $\boldsymbol{n}_j^i, \boldsymbol{m}_j^i \leftarrow$ RK4 using ($\boldsymbol{n}_{j-1}^i, \boldsymbol{m}_{j-1}^i$) and ($\boldsymbol{n}_{s,j-1}^i, \boldsymbol{m}_{s,j-1}^i$)
 6:         $\boldsymbol{v}_j^i, \boldsymbol{u}_j^i \leftarrow$ Forward dynamics (3.8) using $\boldsymbol{n}_j^i, \boldsymbol{m}_j^i$ and $\boldsymbol{v}_{h,j}, \boldsymbol{u}_{h,j}$
 7:         $\boldsymbol{f}_{p,j}^i, \boldsymbol{l}_{p,j}^i \leftarrow$ Control law (3.39) using $\boldsymbol{v}_j^i, \boldsymbol{u}_j^i$
 8:         $\boldsymbol{P}_j^i \leftarrow$ Segment pressure for actual robot from (3.40)
 9:         $\boldsymbol{n}_{s,j}^i, \boldsymbol{m}_{s,j}^i \leftarrow$ Substitute (3.39) in (3.1)
10:     end for
11: end for
12: Note: $(\boldsymbol{n}, \boldsymbol{m}, \boldsymbol{f}, \boldsymbol{l})$ are defined in the global frame, and $(\boldsymbol{v}, \boldsymbol{u}, \boldsymbol{P})$ in the local frame.
```

with respect to arc length, their values at the current spatial node on the backbone, j, are computed (line 5). After implementing the current internal force and moment in (3.8), the values for $\boldsymbol{v}_j^i$ and $\boldsymbol{u}_j^i$ are found using $\boldsymbol{n}_j^i, \boldsymbol{m}_j^i$ from the simulation and $\boldsymbol{v}_{h,j}^i, \boldsymbol{u}_{h,j}^i$ from the actual robot (line 6). Then, using the error between the configuration variables and their desired values, the control law calculates the force and moment ($\boldsymbol{f}_{p,j}^i$,$\boldsymbol{l}_{p,j}^i$) that the actuators must apply to the corresponding backbone section (line 7). The corresponding forces and moments applied by the pneumatic actuators from (3.35) are then mapped to the desired actuator pressures to be sent to the actual robot using (3.40), (3.41), and (3.42) (line 8). Lastly, $\boldsymbol{n}_{s,j}^i$ and $\boldsymbol{m}_{s,j}^i$ are found for the next iteration of the inner loop. (line 9).

3.3.4 Simulation and Experimental Results

In this section, we compare the performance of the controller at tracking desired configurations in both the actual robotic arm and the simulation of the arm, which runs simultaneously in real-time. There are no theoretical limitations imposed by our modeling and decentralized control approach on the number of segments in the robotic arm. However, due to the space limitations for the motion capture system in the laboratory of our collaborator Prof. Wenlong Zhang, we have limited the number of segments in the robotic arm to two segments. Each segment has local sensing, actuation, and control capabilities, and they are therefore independently controllable. The simulated robotic arm is slender, uniform, and symmetric about the z-axis, with isotropic material properties throughout the arm. Although the simulated robot has a circular cross-section (with radius r_0), our control approach can also be applied to multi-segment robots with other cross-section geometries, as long as they satisfy the assumptions required for using the Cosserat rod model.

To demonstrate the performance of the decentralized controller in tracking desired bending and extension configurations, we recorded the position and orientation of the tip of each segment using the motion capture cameras while the controller is implemented. The required control input and outputs are available from both the simulation and experiment. As mentioned earlier, the controller can produce any of the four main deformations; however, due to the robot's physical design restrictions, we are only able to validate the performance of the controller for bending and extension deformations.

In Fig. 3.18, we illustrate the tracking performance of the controller for a case in which both segments experience extension under the following desired reference input:

$$\bar{\boldsymbol{v}}_z(t, s) = 1 + 0.1 \sin^2 (\omega t) \, , \tag{3.43}$$

where $\omega = 2\pi/100$. All the actuators receive the same amount of pressure, which results in almost equal increases in the lengths of both segments. The initial values of L_1 and L_2

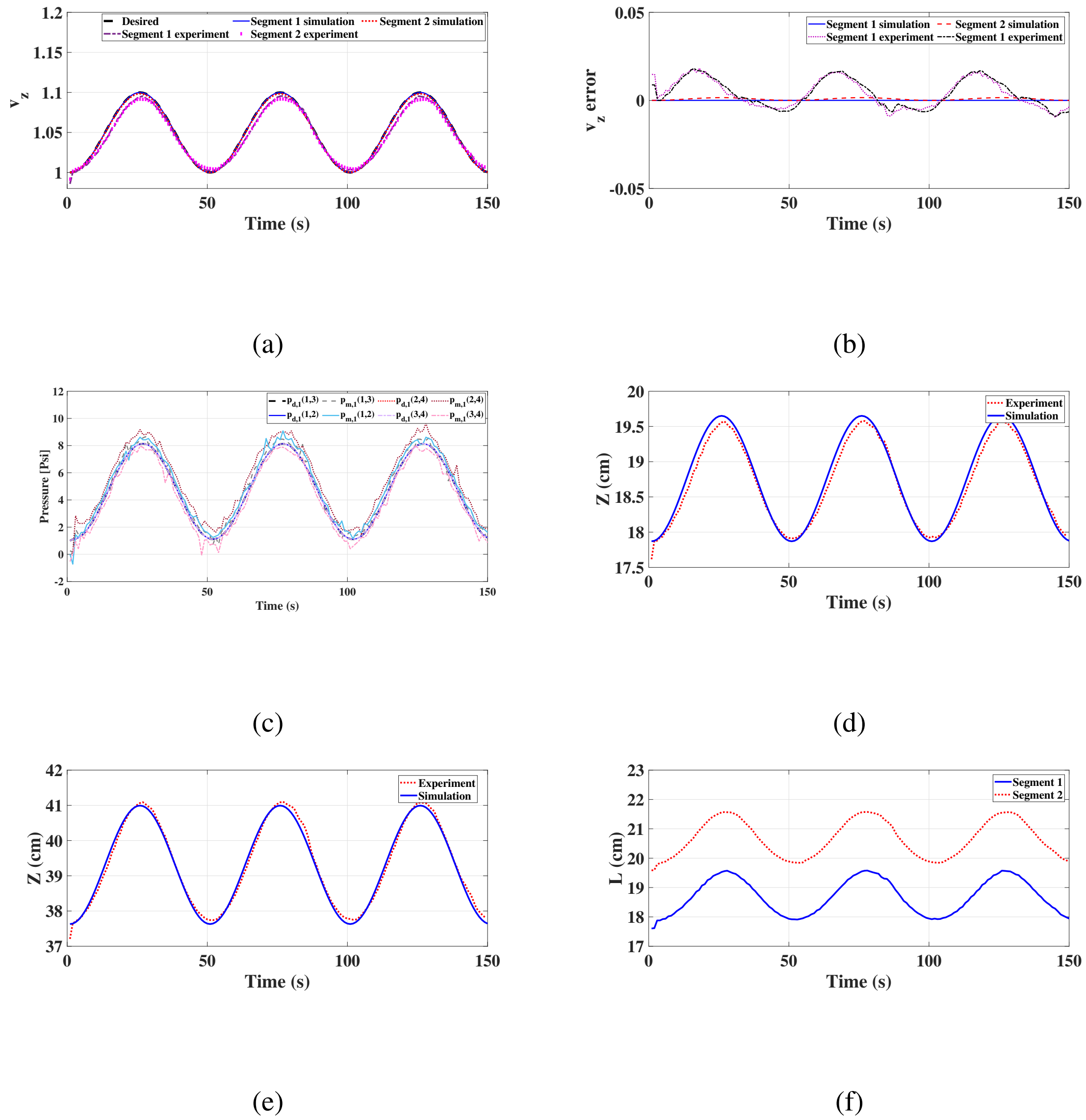
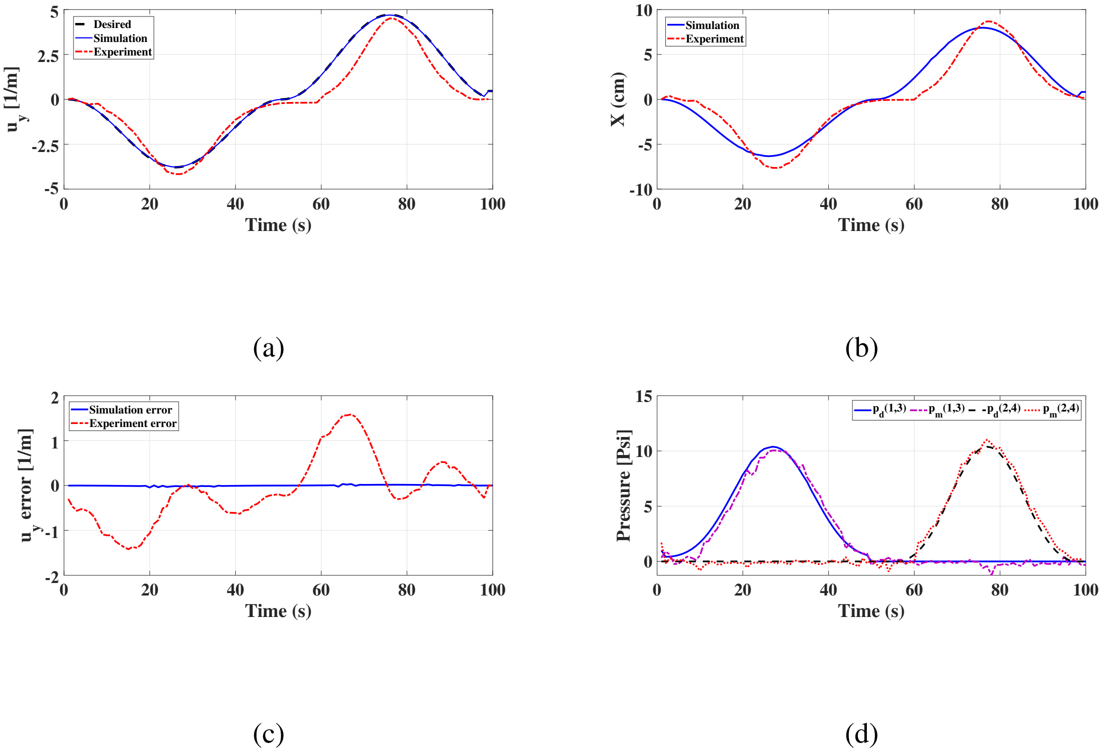


Figure 3.18: Controller performance for tracking a reference extension deformation in the actual multi-segment robotic arm and its simulated model. (a) Component $\boldsymbol{v}_z$ and its corresponding reference, $\bar{\boldsymbol{v}}_z$, for both segments. (b) The error between components $\boldsymbol{v}_z$ and $\boldsymbol{v}_z$. (c) The applied desired and measured pressure, $p_{d,j}(i)$ and $p_{m,j}(i)$ respectively, for actuator i of segment j. (d) Position of segment 1 tip along z-axis. (e) Position of segment 2 tip along z-axis. (f) Length of both segments.



(a)

(b)

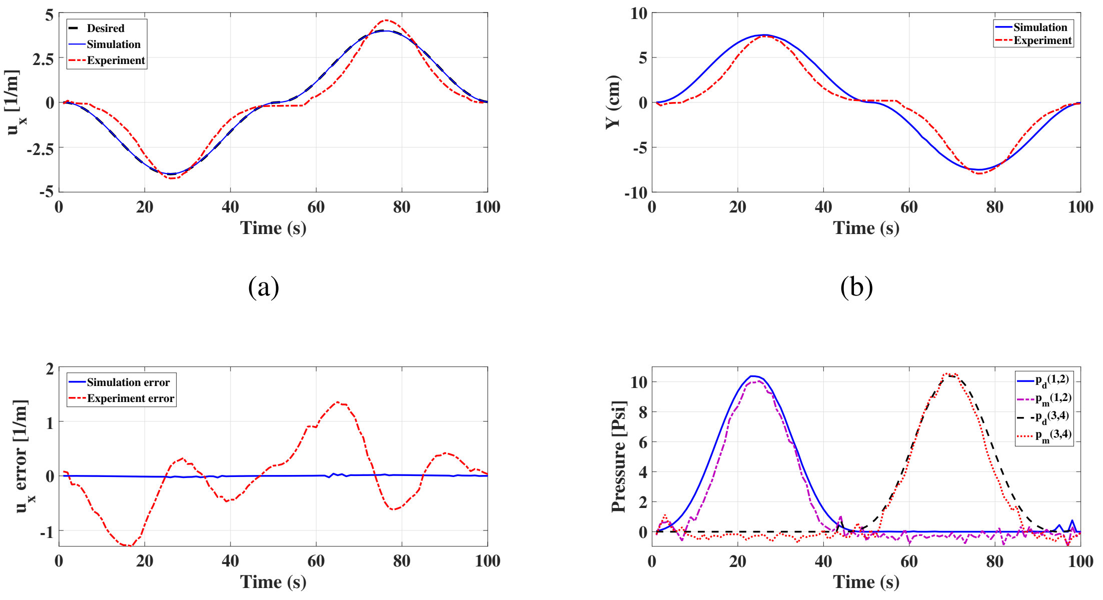
(c)

(d)

Figure 3.19: Controller performance for tracking a reference bending deformation about the y-axis in segment 1 of the robotic arm and its simulated model. (a) Curvature component $\boldsymbol{u}_y$ and its corresponding reference, $\bar{\boldsymbol{u}}_y$. (b) Position of the segment tip. (c) The error between components $\boldsymbol{u}_y$ and $\bar{\boldsymbol{u}}_y$. (d) The applied desired and measured pressure, $p_d(i,j)$ and $p_m(i,j)$, respectively for actuator pairs (i,j).

are slightly different from L_0 due to the locations of the markers on the 3D-printed parts connecting the segments.

Figure 3.19 plots the curvature, error in curvature from the desired reference value, input pressures to the actuators, and x position of segment 1 as it tracks a desired bending deformation about the y-axis. The reference curvature for segment 1 about the y-axis is



(a)

(b)

(c)

(d)

Figure 3.20: Controller performance for tracking a reference bending deformation about the x-axis in segment 2 of the robotic arm and its simulated model. (a) Curvature component $\boldsymbol{u}_x$ and its corresponding reference, $\bar{\boldsymbol{u}}_x$. (b) Position of the segment tip. (c) The error between components $\boldsymbol{u}_x$ and $\bar{\boldsymbol{u}}_x$. (d) The applied desired and measured pressure, $p_d(i,j)$ and $p_m(i,j)$, respectively for actuator pairs (i,j).

defined as: ($\omega = 2\pi/100$)

$$\bar{\boldsymbol{u}}_y(t,s) = \begin{cases} -7.5\sin^2(\omega t) & t \leq 50s, \\ 9.5\sin^2(\omega t) & t > 50s, \end{cases} \tag{3.44}$$

Figure 3.20 plots the curvature, error in curvature from the desired reference value, input pressures to the actuators, and y position of segment 2 as it tracks a desired bending deformation about the x-axis. The reference curvature for segment 2 about the x-axis is defined as: ($\omega = 2\pi/100$)

Table 3.5: NRMSEs From Desired Reference Deformations.

Deformation	Segment	Simulation	Experiment
Extension	1	0.0%	8.8%
Extension	2	0.0%	8.9%
Bending	1	0.1%	8.4%
Bending	2	0.1%	7.7%

$$\bar{\boldsymbol{u}}_x(t,s) = \begin{cases} -8\sin^2(\omega t) & t \leq 50s, \\ 8\sin^2(\omega t) & t > 50s. \end{cases} \tag{3.45}$$

For all three cases, the normalized root-mean-square errors (NRMSEs) were calculated between the simulated and reference configurations, and between the approximated curvature and extension measurements of the actual robot segments and their reference values. These NRMSE values are compared in Table 3.5. The relatively low NRMSEs for the experimental trials (all below 9%) demonstrate the effectiveness of the controller at configuration tracking. The controller's effective performance is also a validation of the accuracy of the dynamic Cosserat rod model of the multi-segment soft robotic arm.

3.3.5 Discussion

In this study, a decentralized control approach is implemented on a pneumatically actuated silicone multi-segment soft robotic arm to track desired configurations in 3D space. The multi-segment robot can perform bending in two directions (about the x and y axes) and elongation in the third direction normal to the other two directions (the z axis). In this control approach, the position and orientation of the tip of each segment are measured using a motion capture system and are fed back to a simulated Cosserat rod model of the robot, which is used to estimate the unmeasured robot parameters and compute the control output. This approach is computationally efficient, unlike many other dynamic controllers, and eliminates the need to equip the robotic arm with additional sensors. The distributed local sensing and actuation of each segment enables the segments to be controlled independently and facilitates tracking of diverse configurations that can potentially be used for complex manipulation tasks.

Chapter 4

CONCLUSION AND FUTURE WORK

In this dissertation, we have taken some essential steps to tackle the challenges of controlling soft robotic arms using model-based approaches, which we hope will be useful for future researchers in soft robotics. We show how decentralized control approaches with local sensing and actuation can be designed to perform complex manipulation and locomotion tasks, drawing on inspiration from invertebrates like octopuses. In recent years, a variety of soft robotic arms with different sources of actuation have been developed; however, the sensing technology required to measure soft robots deflection is still far behind. Therefore, due to experimental and especially sensing limitations, some of the results in this dissertation have only been demonstrated in simulation. The simulated model is developed based on the Cosserat theory of rods, which is capable of reproducing large deformations in slender rods. We have illustrated the accuracy of the simulated model by validating the forward dynamic solution of a miniature hydrogel-based soft robotic arm, including an experimentally identified actuation model. We have also presented a decentralized model-based control approach that is implemented on a pneumatically actuated multi-segment soft robotic arm made of silicone, enabling it to track desired configurations in 3D space.

Complex manipulation tasks such as those performed by octopuses, like opening a jar lid, cannot be accomplished by only controlling the position or shape of the tip of a soft robotic arm; they require control of the configuration of the entire arm throughout the task. Therefore, we believe that our proposed decentralized control approach based on distributed sensing and actuation is a promising starting point for controlling soft robotic arms in applications that require complex manipulation. Defining the desired robot configuration is not trivial for complicated tasks, and hopefully future biological studies and materials developed for soft robots will offer solutions to this problem. In addition, embedded sensors on the robot will be highly useful for controlling its interaction with the environment and can enable autonomous whole-arm grasping (similar to an elephant trunk), which has remained a challenge for soft robotic arms.

REFERENCES

Antman, S. S., “Nonlinear plasticity”, in “Nonlinear Problems of Elasticity”, pp. 603–628 (Springer, 1995).

Antman, S. S., “Problems in nonlinear elasticity”, Nonlinear Problems of Elasticity pp. 513–584 (2005).

Antonelli, G., F. Arrichiello, F. Caccavale and A. Marino, “Decentralized time-varying formation control for multi-robot systems”, The International Journal of Robotics Research **33**, 7, 1029–1043 (2014).

ApS, M., *The MOSEK optimization toolbox for MATLAB manual. Version 8.1.*, URL `http://docs.mosek.com/8.1/toolbox/index.html` (2017).

Åström, K. J. and R. M. Murray, *Feedback systems: an introduction for scientists and engineers* (Princeton University Press, 2010).

Bacelar, A. H., I. F. Cengiz, J. Silva-Correia, R. A. Sousa, J. M. Oliveira and R. L. Reis, ““Smart” hydrogels in tissue engineering and regenerative medicine applications”, in “Handbook of intelligent scaffolds for tissue engineering and regenerative medicine”, pp. 333–367 (Jenny Stanford, 2017).

Bakule, L., “Decentralized control: An overview”, Annual Reviews in Control **32**, 1, 87–98 (2008).

Bayo, E., “Timoshenko versus Bernoulli beam theories for the control of flexible robots”, in “Proceedings of IASTED International Symposium on Applied Control and Identification”, pp. 178–182 (1986).

Behrens, R., 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”, pp. 1–5 (VDE, 2012).

Boyd, S., L. El Ghaoui, E. Feron and V. Balakrishnan, *Linear matrix inequalities in system and control theory*, vol. 15 (SIAM, 1994).

Boyer, F., S. Ali and M. Porez, “Macrocontinuous dynamics for hyperredundant robots: application to kinematic locomotion bioinspired by elongated body animals”, IEEE Transactions on Robotics **28**, 2, 303–317 (2011).

Bugariu, I., S. Micu and I. Rovenţa, “Approximation of the controls for the beam equation with vanishing viscosity”, Mathematics of Computation **85**, 301, 2259–2303 (2016).

Chang, H.-S., U. Halder, E. Gribkova, A. Tekinalp, N. Naughton, M. Gazzola and P. G. Mehta, “Controlling a cyberoctopus soft arm with muscle-like actuation”, arXiv preprint arXiv:2010.03368 (2020a).

Chang, H.-S., U. Halder, C.-H. Shih, A. Tekinalp, T. Parthasarathy, E. Gribkova, G. Chowdhary, R. Gillette, M. Gazzola and P. G. Mehta, “Energy shaping control of a cyberoctopus soft arm”, arXiv preprint arXiv:2004.05747 (2020b).

Chirikjian, G. S., “Hyper-redundant manipulator dynamics: A continuum approximation”, Advanced Robotics **9**, 3, 217–243 (1994).

Cianchetti, M., A. Arienti, M. Follador, B. Mazzolai, P. Dario and C. Laschi, “Design concept and validation of a robotic arm inspired by the octopus”, Materials Science and Engineering: C **31**, 6, 1230–1239 (2011).

Della Santina, C., R. K. Katzschmann, A. Bicchi and D. Rus, “Model-based dynamic feedback control of a planar soft robot: Trajectory tracking and interaction with the environment”, The International Journal of Robotics Research **39**, 4, 490–513 (2020).

Doroudchi, A. and S. Berman, “Configuration tracking for soft continuum robotic arms using inverse dynamic control of a Cosserat rod model”, IEEE International Conference on Soft Robotics (RoboSoft) (2021).

Doroudchi, A., R. Khodambashi, A. S. Lafmejani, D. M. Aukes and S. Berman, “Dynamic modeling of a hydrogel-based continuum robotic arm with experimental validation”, in “IEEE Int’l. Conference on Soft Robotics (RoboSoft)”, pp. 695–701 (2020).

Doroudchi, A., R. Khodambashi, M. Sharifzadeh, D. Li, S. Berman and D. M. Aukes, “Tracking control of a miniature 2-dof manipulator with hydrogel actuators”, IEEE Robotics and Automation Letters (RA-L) (2021).

Doroudchi, A., Z. Qiao, W. Zhang and S. Berman, “Implementation of a decentralized tracking controller on a silicone-based soft robotic arm using Cosserat rod dynamics”, In preparation for IEEE Robotics and Automation Letters (RA-L) (2022).

Doroudchi, A., 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 “2018 IEEE Conference on Decision and Control (CDC)”, pp. 7002–7009 (IEEE, 2018).

Duan, G.-R. and H.-H. Yu, *LMIs in control systems: analysis, design and applications* (CRC Press, 2013).

Duriez, C., “Control of elastic soft robots based on real-time finite element method”, in “IEEE Int’l. Conference on Robotics and Automation (ICRA)”, pp. 3982–3987 (2013).

Farhamfard, A., M. B. Menhaj and A. Fakharian, “H_∞ output feedback controller design for flexible needles guidance”, in “2016 Artificial Intelligence and Robotics (IRANOPEN)”, pp. 164–169 (IEEE, 2016).

George Thuruthel, T., Y. Ansari, E. Falotico and C. Laschi, “Control strategies for soft robotic manipulators: A survey”, Soft Robotics **5**, 2, 149–163 (2018).

George Thuruthel, T., F. Renda and F. Iida, “First-order dynamic modeling and control of soft robots”, Frontiers in Robotics and AI **7**, 95 (2020).

Godage, I. S., G. A. Medrano-Cerda, D. T. Branson, E. Guglielmino and D. G. Caldwell, “Modal kinematics for multisection continuum arms”, Bioinspiration & Biomimetics **10**, 3, 035002 (2015).

Godage, I. S., G. A. Medrano-Cerda, D. T. Branson, E. Guglielmino and D. G. Caldwell, “Dynamics for variable length multisection continuum arms”, The International Journal of Robotics Research **35**, 6, 695–722 (2016).

Goh, K.-C., M. G. Safonov and G. P. Papavassilopoulos, “A global optimization approach for the BMI problem”, in “Proceedings of the 33rd Conference on Decision and Control (CDC)”, vol. 3, pp. 2009–2014 (IEEE, 1994).

Grasso, F. W., “The octopus with two brains: how are distributed and central representations integrated in the octopus central nervous system”, Cephalopod cognition pp. 94–122 (2014).

Gravagne, I. A., C. D. Rahn and I. D. Walker, “Large deflection dynamics and control for planar continuum robots”, IEEE/ASME transactions on mechatronics **8**, 2, 299–307 (2003).

Gravagne, I. A. and I. D. Walker, “Uniform regulation of a multi-section continuum manipulator”, in “Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)”, vol. 2, pp. 1519–1524 (IEEE, 2002).

Guan, Q., J. Sun, Y. Liu, N. M. Wereley and J. Leng, “Novel bending and helical extensile/contractile pneumatic artificial muscles inspired by elephant trunk”, Soft Robotics (2020).

Guiseppi-Elie, A., “Electroconductive hydrogels: synthesis, characterization and biomedical applications”, Biomaterials **31**, 10, 2701–2716 (2010).

Gutfreund, Y., H. Matzner, T. Flash and B. Hochner, “Patterns of motor activity in the isolated nerve cord of the octopus arm”, The Biological Bulletin **211**, 3, 212–222 (2006).

Han, D., C. Farino, C. Yang, T. Scott, D. Browe, W. Choi, J. W. Freeman and H. Lee, “Soft Robotic Manipulation and Locomotion with a 3D Printed Electroactive Hydrogel”, ACS Applied Materials & Interfaces **10**, 17512–17518 (2018).

Hassibi, A., J. How and S. Boyd, “A path-following method for solving BMI problems in control”, in “Proceedings of the American Control Conference (ACC)”, vol. 2, pp. 1385–1389 (IEEE, 1999).

He, X., M. Aizenberg, O. Kuksenok, L. D. Zarzar, A. Shastri, A. C. Balazs and J. Aizenberg, “Synthetic homeostatic materials with chemo-mechano-chemical self-regulation”, Nature **487**, 7406, 214 (2012).

Hines, L., K. Petersen, G. Z. Lum and M. Sitti, “Soft Actuators for Small-Scale Robotics”, Advanced Materials **29**, 13 (2017).

Holsapple, R. W., *A modified simple shooting method for solving two-point boundary value problems*, Ph.D. thesis, Texas Tech University (2003).

Hong, W., Z. Liu and Z. Suo, “Inhomogeneous swelling of a gel in equilibrium with a solvent and mechanical load”, International Journal of Solids and Structures **46**, 17, 3282–3289 (2009).

Hong, W., X. Zhao, J. Zhou and Z. Suo, “A theory of coupled diffusion and large deformation in polymeric gels”, Journal of the Mechanics and Physics of Solids **56**, 5, 1779–1793 (2008).

Ibaraki, S. and M. Tomizuka, “Rank minimization approach for solving BMI problems with random search”, in “Proceedings of the American Control Conference (ACC)”, vol. 3, pp. 1870–1875 (IEEE, 2001).

Ionov, L., “Hydrogel-based actuators: Possibilities and limitations”, Materials Today **17**, 10, 494–503, URL `http://dx.doi.org/10.1016/j.mattod.2014.07.002` (2014).

Ishimura, K., 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”, pp. 227–236 (Springer, 2002).

Ivanescu, M., N. Popescu and D. Popescu, “The shape control of tentacle arms”, Robotica **33**, 3, 684–703 (2015).

Jiang, H., Z. Wang, Y. Jin, X. Chen, P. Li, Y. Gan, S. Lin and X. Chen, “Design, control, and applications of a soft robotic arm”, arXiv preprint arXiv:2007.04047 (2020).

Jiang, H., Z. Wang, Y. Jin, X. Chen, P. Li, Y. Gan, S. Lin and X. Chen, “Hierarchical control of soft manipulators towards unstructured interactions”, The International Journal of Robotics Research **40**, 1, 411–434 (2021).

Kanev, S., C. Scherer, M. Verhaegen and B. De Schutter, “Robust output-feedback controller design via local BMI optimization”, Automatica **40**, 7, 1115–1127 (2004).

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

Kapadia, A. and I. D. Walker, “Task-space control of extensible continuum manipulators”, in “IEEE/RSJ Int’l. Conference on Intelligent Robots and Systems (IROS)”, pp. 1087–1092 (2011).

Katzschmann, R. K., M. Thieffry, O. Goury, A. Kruszewski, T.-M. Guerra, C. Duriez and D. Rus, “Dynamically closed-loop controlled soft robotic arm using a reduced order finite element model with state observer”, in “IEEE Int’l. Conference on Soft Robotics (RoboSoft)”, pp. 717–724 (2019).

Kerkut, G., “The forces exerted by the tube feet of the starfish during locomotion”, Journal of Experimental Biology **30**, 4, 575–583 (1953).

Khalil, H. K. and J. W. Grizzle, *Nonlinear systems*, vol. 3 (Prentice Hall Upper Saddle River, NJ, 2002).

Khodambashi, R., Y. Alsaid, D. Aukes and X. He, “Shape Morphing Soft Material”, US Provisional Patent 62/860,700 (2019).

Khodambashi, R., Y. Alsaid, R. Rico, H. Marvi, M. M. Peet, R. E. Fisher, S. Berman, X. He and D. M. Aukes, "Heterogeneous hydrogel structures with spatiotemporal reconfigurability using addressable and tunable voxels", Advanced Materials **33**, 10, 2005906 (2021).

Kier, W. M., "The arrangement and function of molluscan muscle", The mollusca, form and function **11**, 21, 211–252 (1988).

Kier, W. M., "The musculature of coleoid cephalopod arms and tentacles", Frontiers in cell and developmental biology **4**, 10 (2016).

Kier, W. M. and K. K. Smith, "Tongues, tentacles and trunks: the biomechanics of movement in muscular-hydrostats", Zoological Journal of the Linnean Society **83**, 4, 307–324 (1985).

Kim, S., C. Laschi and B. Trimmer, "Soft robotics: a bioinspired evolution in robotics", Trends in Biotechnology **31**, 5, 287–294 (2013).

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

Kolachalama, S. and S. Lakshmanan, "Continuum robots for manipulation applications: A survey", Journal of Robotics **2020** (2020).

Kwon, G. H., J. Y. Park, J. Y. Kim, M. L. Frisk, D. J. Beebe and S.-H. Lee, "Biomimetic soft multifunctional miniature aquabots", Small **4**, 12, 2148–2153 (2008).

Lafmejani, A. S., 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 **5**, 2, 3460–3467 (2020a).

Lafmejani, A. S., H. Farivarnejad, A. Doroudchi and S. Berman, "A consensus strategy for decentralized kinematic control of multi-segment soft continuum robots", in "American Control Conference (ACC)", pp. 909–916 (2020b).

Lang, H., J. Linn and M. Arnold, "Multi-body dynamics simulation of geometrically exact cosserat rods", Multibody System Dynamics **25**, 3, 285–312 (2011).

Laschi, C. and M. Cianchetti, "Soft robotics: new perspectives for robot bodyware and control", Frontiers in Bioengineering and Biotechnology **2**, 3 (2014).

Lessard, L. and S. Lall, "Quadratic invariance is necessary and sufficient for convexity", in "Proceedings of the American Control Conference (ACC)", pp. 5360–5362 (IEEE, 2011).

Li, H., "Smart hydrogel modelling", Springer (2009).

Lim, R. K., M. Q. Phan and R. W. Longman, “State-space system identification with identified Hankel matrix”, Department of Mechanical and Aerospace Engineering Technical Report **3045** (1998).

Linn, J., H. Lang and A. Tuganov, “Geometrically exact cosserat rods with kelvin–voigt type viscous damping”, Mechanical Sciences **4**, 1, 79–96 (2013).

Liu, S.-M. and G. P. Papavassilopoulos, “Numerical experience with parallel algorithms for solving the BMI problem”, IFAC Proceedings Volumes **29**, 1, 1827–1832 (1996).

Liu, Y.-H. and S. Arimoto, “Decentralized adaptive and nonadaptive position/force controllers for redundant manipulators in cooperations”, The International Journal of Robotics Research **17**, 3, 232–247 (1998).

Ljung, L., “Black-box models from input-output measurements”, in “Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference (IMTC 2001). Rediscovering Measurement in the Age of Informatics (Cat. No. 01CH 37188)”, vol. 1, pp. 138–146 (IEEE, 2001a).

Ljung, L., “Black-box models from input-output measurements”, in “IMTC 2001. Proceedings of the 18th IEEE instrumentation and measurement technology conference. Rediscovering measurement in the age of informatics (Cat. No. 01CH 37188)”, vol. 1, pp. 138–146 (IEEE, 2001b).

Ljung, L., “Perspectives on system identification”, Annual Reviews in Control **34**, 1, 1–12 (2010).

Lofberg, J., “YALMIP: a toolbox for modeling and optimization in MATLAB”, in “Proceedings of International Symposium on Computer Aided Control Systems Design”, pp. 284–289 (IEEE, 2004).

Majidi, C., “Soft robotics: a perspective—current trends and prospects for the future”, Soft Robotics **1**, 1, 5–11 (2014).

Margheri, L., B. Mazzolai, M. Cianchetti, P. Dario and C. Laschi, “Tools and methods for experimental in-vivo measurement and biomechanical characterization of an octopus vulgaris arm”, in “2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society”, pp. 7196–7199 (IEEE, 2009).

Matsuno, F. and K. Suenaga, “Control of redundant 3D snake robot based on kinematic model”, in “Proceedings of International Conference on Robotics and Automation (ICRA)”, vol. 2, pp. 2061–2066 (IEEE, 2003).

McEvoy, M. A. and N. Correll, “Shape-changing materials using variable stiffness and distributed control”, Soft robotics **5**, 6, 737–747 (2018).

Mishra, A. K., T. J. Wallin, W. Pan, P. Xu, K. Wang, E. P. Giannelis, B. Mazzolai and R. F. Shepherd, “Autonomic perspiration in 3d-printed hydrogel actuators”, Science Robotics **5**, 38 (2020).

Morales Bieze, T., A. Kruszewski, B. Carrez and C. Duriez, "Design, implementation, and control of a deformable manipulator robot based on a compliant spine", The International Journal of Robotics Research **39**, 14, 1604–1619 (2020).

Naughton, N., J. Sun, A. Tekinalp, G. Chowdhary and M. Gazzola, "Elastica: A compliant mechanics environment for soft robotic control", arXiv preprint arXiv:2009.08422 (2020).

Nguyen, P. H., S. Sridar, W. Zhang and P. Polygerinos, "Design and control of a 3-chambered fiber reinforced soft actuator with off-the-shelf stretch sensors", International Journal of Intelligent Robotics and Applications **1**, 3, 342–351 (2017).

Niu, L., L. Ding, H. Gao, Y. Su, Z. Deng and Z. Liu, "Closed-form equations and experimental verification for soft robot arm based on Cosserat theory", in "IEEE/RSJ Int'l. Conference on Intelligent Robots and Systems (IROS)", pp. 6630–6635 (2019).

O'Reilly, O. M., *Modeling nonlinear problems in the mechanics of strings and rods* (Springer, 2017).

Peng, X., C. Jiao, Y. Zhao, N. Chen, Y. Wu, T. Liu and H. Wang, "Thermoresponsive deformable actuators prepared by local electrochemical reduction of poly (n-isopropylacrylamide)/graphene oxide hydrogels", ACS Applied Nano Materials **1**, 4, 1522–1530 (2018).

Pentreath, R., "Feeding mechanisms and the functional morphology of podia and spines in some New Zealand ophiuroids (Echinodermata)", Journal of Zoology **161**, 3, 395–429 (1970).

Pfeifer, R., M. Lungarella and F. Iida, "The challenges ahead for bio-inspired'soft'robotics", Communications of the ACM **55**, 11, 76–87 (2012).

Qin, M., M. Sun, M. Hua and X. He, "Bioinspired structural color sensors based on responsive soft materials", Current Opinion in Solid State and Materials Science **23**, 1, 13–27 (2019).

Renda, F., V. Cacucciolo, J. Dias and L. Seneviratne, "Discrete cosserat approach for soft robot dynamics: A new piece-wise constant strain model with torsion and shears", in "2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)", pp. 5495–5502 (IEEE, 2016).

Renda, F. and C. Laschi, "A general mechanical model for tendon-driven continuum manipulators", in "2012 IEEE International Conference on Robotics and Automation", pp. 3813–3818 (IEEE, 2012).

Richter, J. N., B. Hochner and M. J. Kuba, "Octopus arm movements under constrained conditions: adaptation, modification and plasticity of motor primitives", Journal of Experimental Biology **218**, 7, 1069–1076 (2015).

Rotkowitz, M., R. Cogill and S. Lall, "Convexity of optimal control over networks with delays and arbitrary topology", International Journal of Systems, Control and Communications **2**, 1-3, 30–54 (2010).

Rucker, D. C. and R. J. Webster III, "Statics and dynamics of continuum robots with general tendon routing and external loading", IEEE Transactions on Robotics **27**, 6, 1033–1044 (2011).

Rus, D. and M. T. Tolley, "Design, fabrication and control of soft robots", Nature **521**, 7553, 467 (2015).

Safonov, M. G., K.-C. Goh and J. H. Ly, "Control system synthesis via bilinear matrix inequalities", in "Proceedings of the American Control Conference (ACC)", vol. 1, pp. 45–49 (IEEE, 1994).

Schild, H. G., "Poly (n-isopropylacrylamide): experiment, theory and application", Progress in polymer science **17**, 2, 163–249 (1992).

Siciliano, B., L. Sciavicco, L. Villani and G. Oriolo, *Robotics: modelling, planning and control* (Springer Science & Business Media, 2010).

Siljak, D. D., *Decentralized control of complex systems* (Courier Corporation, 2011).

Snyder, J. and J. Wilson, "Dynamics of the elastica with end mass and follower loading", Journal of applied mechanics **57**, 1, 203–208 (1990).

Spillmann, J. and M. Teschner, "Corde: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects", in "Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation", pp. 63–72 (Eurographics Association, 2007).

Stipanović, D. M., G. Inalhan, R. Teo and C. J. Tomlin, "Decentralized overlapping control of a formation of unmanned aerial vehicles", Automatica **40**, 8, 1285–1296 (2004).

Stoer, J. and R. Bulirsch, *Introduction to numerical analysis*, vol. 12 (Springer Science & Business Media, 2013).

Tatlicioglu, E., I. D. Walker and D. M. Dawson, "Dynamic modelling for planar extensible continuum robot manipulators", in "Proceedings 2007 IEEE International Conference on Robotics and Automation", pp. 1357–1362 (IEEE, 2007).

Till, J., V. Aloi and C. Rucker, "Real-time dynamics of soft and continuum robots based on cosserat rod models", The International Journal of Robotics Research **38**, 6, 723–746 (2019).

Till, J., C. E. Bryson, S. Chung, A. Orekhov and D. C. Rucker, "Efficient computation of multiple coupled cosserat rod models for real-time simulation and control of parallel continuum manipulators", in "2015 IEEE International Conference on Robotics and Automation (ICRA)", pp. 5067–5074 (IEEE, 2015).

Till, J. and D. C. Rucker, "Elastic stability of cosserat rods and parallel continuum robots", IEEE Transactions on Robotics **33**, 3, 718–733 (2017).

Toker, O. and H. Ozbay, "On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback", in "Proceedings of the American Control Conference (ACC)", vol. 4, pp. 2525–2526 (IEEE, 1995).

Trivedi, D., A. Lotfi and C. D. Rahn, “Geometrically exact models for soft robotic manipulators”, IEEE Transactions on Robotics **24**, 4, 773–780 (2008a).

Trivedi, D., C. D. Rahn, W. M. Kier and I. D. Walker, “Soft robotics: Biological inspiration, state of the art, and future research”, Applied bionics and biomechanics **5**, 3, 99–117 (2008b).

Tuan, H. D., P. Apkarian and Y. Nakashima, “A new Lagrangian dual global optimization algorithm for solving bilinear matrix inequalities”, International Journal of Robust and Nonlinear Control **10**, 7, 561–578 (2000).

Umedachi, T. and B. A. Trimmer, “Autonomous decentralized control for soft-bodied caterpillar-like modular robot exploiting large and continuum deformation”, in “2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)”, pp. 292–297 (IEEE, 2016).

VanAntwerp, J. G. and R. D. Braatz, “A tutorial on linear and bilinear matrix inequalities”, Journal of Process Control **10**, 4, 363–385 (2000).

VanAntwerp, J. G., R. D. Braatz and N. V. Sahinidis, “Globally optimal robust control for systems with time-varying nonlinear perturbations”, Computers & Chemical Engineering **21**, S125–S130 (1997).

VanAntwerp, J. G., R. D. Braatz and N. V. Sahinidis, “Globally optimal robust process control”, Journal of Process Control **9**, 5, 375–383 (1999).

Walker, I. D., “Continuous backbone “continuum” robot manipulators”, ISRN Robotics Volume 2013 (2013a).

Walker, I. D., “Continuous backbone “continuum” robot manipulators”, Isrn robotics **2013** (2013b).

Walker, I. D., H. Choset and G. S. Chirikjian, “Snake-like and continuum robots”, in “Springer Handbook of Robotics”, pp. 481–498 (Springer, 2016).

Wang, C., C. G. Frazelle, J. R. Wagner and I. Walker, “Dynamic control of multi-section three-dimensional continuum manipulators based on virtual discrete-jointed robot models”, IEEE/ASME Transactions on Mechatronics (2020).

Wang, T., J. Huang, Y. Yang, E. Zhang, W. Sun and Z. Tong, “Bioinspired Smart Actuator Based on Graphene Oxide-Polymer Hybrid Hydrogels”, ACS Applied Materials and Interfaces **7**, 42, 23423–23430 (2015).

Wang, X., Z. Zhai, Y. Chen and H. Jiang, “A facile, robust and versatile finite element implementation to study the time-dependent behaviors of responsive gels”, Extreme Mechanics Letters **22**, 89–97 (2018).

Webster III, R. J. and B. A. Jones, “Design and kinematic modeling of constant curvature continuum robots: A review”, The International Journal of Robotics Research **29**, 13, 1661–1683 (2010a).

Webster III, R. J. and B. A. Jones, “Design and kinematic modeling of constant curvature continuum robots: A review”, The International Journal of Robotics Research **29**, 13, 1661–1683 (2010b).

Webster III, R. J. and B. A. Jones, “Design and kinematic modeling of constant curvature continuum robots: A review”, The International Journal of Robotics Research **29**, 13, 1661–1683 (2010c).

Webster III, R. J., J. M. Romano and N. J. Cowan, “Mechanics of precurved-tube continuum robots”, IEEE Transactions on Robotics **25**, 1, 67–78 (2008).

Xia, L.-W., R. Xie, X.-J. Ju, W. Wang, Q. Chen and L.-Y. Chu, “Nano-structured smart hydrogels with rapid response and high elasticity”, Nature communications **4**, 2226 (2013).

Yamada, Y. and S. Hara, “Global optimization for H_∞ control with constant diagonal scaling”, IEEE Transactions on Automatic Control **43**, 2, 191–203 (1998).

Yeghiazarian, B. L., S. Mahajan, C. Montemagno, C. Cohen and U. Wiesner, “Directed Motion and Cargo Transport Through Propagation of Polymer-Gel Volume Phase Transitions”, Advanced Materials **17**, 15, 1869–1873 (2005).

Yekutieli, Y., R. Sagiv-Zohar, R. Aharonov, Y. Engel, B. Hochner and T. Flash, “Dynamic model of the octopus arm. i. biomechanics of the octopus reaching movement”, Journal of neurophysiology **94**, 2, 1443–1458 (2005a).

Yekutieli, Y., R. Sagiv-Zohar, B. Hochner and T. Flash, “Dynamic model of the octopus arm. ii. control of reaching movements”, Journal of neurophysiology **94**, 2, 1459–1468 (2005b).

Yu, C., Z. Duan, P. Yuan, Y. Li, Y. Su, X. Zhang, Y. Pan, L. L. Dai, R. G. Nuzzo, Y. Huang, H. Jiang and J. A. Rogers, “Electronically programmable, reversible shape change in two- and three-dimensional hydrogel structures”, Advanced Materials **25**, 11, 1541–1546 (2013).

Zolfagharian, A., A. Z. Kouzani, S. Y. Khoo, A. Noshadi and A. Kaynak, “3d printed soft parallel actuator”, Smart Materials and Structures **27**, 4, 045019 (2018).