UNIVERSITY OF CALIFORNIA
SANTA CRUZ

STUDY OF KINESTHETIC FEEDBACK CONTROL FOR COMPLIANT PROPRIOCEPTIVE TOUCH FOR SOFT ROBOTIC FINGER-LIKE ACTUATORS

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING
with an emphasis in ROBOTICS AND CONTROL

by

Megan Boivin

June 2021

The Dissertation of Megan Boivin
is approved:

Professor Dejan Milutinović, Chair

Professor Michael Wehner

Professor Ricardo Sanfelice

Quentin Williams
Acting Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

Abstract

Study of Kinesthetic Feedback Control for Compliant Proprioceptive Touch for Soft Robotic Finger-like Actuators

by

Megan Boivin

The compliant nature of soft robotic components lends itself well to manipulation and contact-rich tasks. The soft structures naturally form around objects where there could be uncertainty in shape or orientation and are inherently safer for fragile objects and humans. However, this compliancy makes the robot's movement less constrained and less predictable making it difficult to control the position of the soft robotic manipulator without new types of sensing. To address these drawbacks, the combination of curvature, inflation, and contact sensors are added to give the finger the unique capability of somatosensory abilities, enhancing how it interacts with objects and its controllability. In this dissertation, we present the application of various approaches that provide a sense of touch for compliant soft robotic fingers. We rethink and validate the role that sensory feedback can play in the control of soft finger-like actuators with proprioceptive sensing capabilities. In our method of touch detection, instead of using the sensory feedback to control precisely the position of a soft finger, we use the disagreement between the controlled curvature sensor measurement and its reference signal to detect the contact between the soft finger and an object. We first consider the case of a static characteristic relation between the inflation pressure and sensor resistance. A control architecture is presented utilizing both the curvature and force sensors with the aim of providing a firm touch of a soft somatosensitive

actuator with an object. The first component of the architecture, a reference tracking curvature controller, sets the finger in motion, which becomes blocked if an object is in its path. The result of such an event is that the finger bending is constrained, and the tracking error of the curvature controller increases. Once the error exceeds a predetermined threshold value, there is a switch from the reference tracking curvature controller to the second component, a force controller, which maintains the finger in contact with the object for a certain pressure using the force sensor measurement. We next consider a method for a kinesthetic touch approach for object detection that does not require the force sensor; therefore, it overcomes the necessity for the co-location between the point of contact and the corresponding sensor. The control architecture uses only the finger's proprioceptive curvature sensor to detect contact with an object and maintain contact by switching to a different reference to hold at a constant curvature. Lastly, we focus on a method for identifying the dynamic finger curvature model to improve the closed-loop control. The proposed method addresses environmental variations as well as variations in material or human factors during the fabrication process that can have an effect on the finger dynamics. The approach uses the reference tracking error as a measure of the finger stress resulting from a contact with an object. The approach for contact detection has been tested in various tasks, including keeping the finger in a firm touch with an object, detecting the object edges and visualizing the operating space based on the sense of touch. These tasks demonstrate that the error signal contains robust information regarding the finger's sense of touch and how it interacts with the environment. Findings are demonstrated across both a proxy of a soft finger, a simulated compliant multi-link actuator with flexible joints, and a real soft robotic finger.

Acknowledgments

The completion of this thesis would not have been possible without the support and encouragement of several special people. Hence, I would like to take this opportunity to show my gratitude for the help and support of many individuals in reaching this milestone.

A close friend often reminded me in times of frustration during graduate school that everyone is on her own unique academic journey. First and foremost, I would like to express my heartfelt thanks to my advisor, Dr.-Ing. Dejan Milutinović, who supported me and my research throughout my unique journey, with patience and kindness. I am deeply grateful for the opportunity to have learned from him; the enjoyment and enthusiasm he brings to research is truly inspiring. His willingness to offer me so much of his time and intellect is a major reason this thesis was completed. I would also like to extend my sincere appreciation to my co-advisor, Dr. Michael Wehner, for his support and guidance during the last few years of my dissertation work. I am forever grateful for the collaboration that has formed the body of this work. Thank you to both of my advisors for your continued encouragement and guidance in pursuing research during my PhD program and beyond. With both of your mentorship I feel prepared for new challenges in the future. Next I would like to thank Dr. Ricardo Sanfelice for participating in my Ph.D. dissertation committee and for his time and interest in evaluating my work. Many thanks to my colleagues and collaborators; your efforts were crucial to the completion of this work. To Keng-yu Lin, thank you for your generosity in providing your time to help me complete the remainder of the experiments during these challenging and uncertain times. To Conrad Esch, thank you for your attention to detail and enthusiasm in designing the finger presented in this

work. To Rory Grant, thank you for helping us with planning for the robot arm motion in V-REP. To my former and current labmates, thank you for the camaraderie, encouragement, ideas and for sharing in both the good and the bad days.

Finally, an enormous thank you to my family and friends. I would not have made it here were it not for your support. I owe the greatest debt of gratitude to my late mother, Sandy Kabureck, and my father, James Boivin. Every step and stumble I've taken has been met with love and support, and I thank you for instilling in me the drive to keep going that has allowed me to reach this milestone. Pursing something as demanding as a graduate degree needs at least some play for balance, and I must thank my friends for always helping me to get away and take a break. Thank you for reminding me of what is important in life.

This research was partially supported by a 2019 Seed Fund Award from CITRIS and the Banatao Institute at the University of California, and was partially supported by the Hellman Fellows Program to Professor Michael Wehner. Their support is gratefully acknowledged.

A Note on Previously Published Material

The text of this dissertation includes reprint of the following material:

- M. Boivin, D. Milutinović, and M. Wehner, "Movement Error Based Control for a Firm Touch of a Soft Somatosensitive Actuator," in 2019 American Control Conference (ACC). IEEE, 2019, pp. 7-12.

With regards to the authors of these reprints, the co-author Dejan Milutinović (dmilutin@ucsc.edu) and Michael Wehner (miwehner@ucsc.edu) listed in these publications directed and supervised

the research contained within this dissertation.

"You may encounter many defeats, but you must not be defeated. In fact, it may be necessary to encounter the defeats, so you can know who you are, what you can rise from, how you can still come out of it."

– Maya Angelou

Dedicated to the memory of my mother, Sandy, and mémère, two of the most influential women in my life who never saw this adventure completed. They were a great inspiration to me. Thank you for always encouraging me to try new things and reminding me it's alright to make mistakes as long as we learn from them and continue to grow. I am forever grateful for all of their love, support, and instilling in me a sense of determination that allowed me to keep growing and learning.

Chapter 1

Introduction

1.1 Motivation

Conventional rigid-bodied robots are excellent at performing well defined repetitive tasks in highly structured environments where they deal with contact that has been planned such as assembly lines in the manufacturing industry. However, they are not as good at dealing with uncertainty in contact in unstructured environments that we experience in every-day life and can pose safety hazards to humans in cooperative tasks due to the amount of force they can exert from their rigid structure and the lack of inherent compliance and adaptability. There has been some work in designing force controllers for rigid manipulators where the focus is on the force position relation from how the robot interacts with its environment [2, 53] that improves some of the safety hazards, but does not address unstructured environments. The desire to design robots to interact with humans and in unstructured spaces where contact with the environment is necessary has encouraged the robotics community to invent new pathways, among which is soft

robotics.

Inspired by soft flexible manipulation often found in nature [64, 79], soft robots that are designed to be compliant either using flexible joints or composed of soft materials are an attractive alternative to rigid manipulators [67, 105]. Their inherent compliance reduces the harm that could be inadvertently caused by traditional rigid robotic systems for human-centric environments as well as handling delicate objects where safety and adaptability to uncertainties are fundamental requirements [98, 105]. There are many areas and applications in which soft robots are well suited for example the biomedical industry [25] including: surgical [18], prosthetics [97, 129], and exoskeletons [70, 100, 125]; search and rescue [51, 115]; deep sea exploration [41, 72]; and the agricultural industry [54].

Many tasks in robotics require physical interactions between the robot and its environment. Due to their compliant nature, soft robots not only improve safety concerns for human interaction but can also simplify some contact tasks such as grasping, by adaptively conforming to objects where there could be uncertainty in shape and orientation as well as handling fragile objects making them extremely versatile [14]. However, due to the flexibility, soft robotic components have less constrained motion that are challenging to model and control [42, 105]. The motion is even more unpredictable when the component makes contact with objects in the operating environment. To deal with the complexity of its motion we want to infer the configuration of the component at a given moment using advanced internal sensing approaches like proprioception. As sensor technology advances, robotic manipulators are being fit with sensors [71] at the finger tips for detecting contact with objects. Soft robotic components are among this category that are becoming increasingly popular, and benefit from sensory feedback

for knowledge about their shape for closed-loop control. Embedding stretchable, distributed sensors [24] in or on soft robotic components improves knowledge about the robot's configuration and how they interact with objects [110]. In addition, this also increases how they can be controlled for both physics-based and data-driven models.

Even with advanced sensing, there are still challenges in robot operations related to interactions with objects of unknown location, shape, orientation and surface characteristics. Force sensors at the tip of the fingers are sensitive to the collocation between the sensor and point of contact on the object to trigger contact detection reducing the effectiveness in real world scenarios. To address these challenges, many contact-dependent tasks combine multiple sensing modalities from vision and touch, and manipulators rely on processing visual data from cameras and detecting contact through dedicated force sensors. However, the information from vision modalities could become unreliable if line of sight becomes blocked, the light source fails [68], or when dealing with transparent objects [57, 107]. In these situations, humans have learned how to adapt when vision becomes unreliable by using the sense of touch through the somatosensory system of their bodies. Inspired by the human sensory system, our work focuses on the design of a dynamically based kinesthetic feedback control architecture for a finger-like soft robotic component with embedded sensors to enable touch through proprioceptive sensing. By enabling the robot to have a robust sense of touch at the end-effector the robot's workspace can still be visualized when combining that data with image processing techniques without needing to rely on precise vision data or contact location at the finger tip.

In humans, the sense of touch is described by two modalities, *cutaneous* and *kinesthetic* [27]. The cutaneous sense of touch refers to the modality that is often thought of as haptic or

tactile sensing where the sensory input is from the receptors embedded in the skin, while the kinesthetic sense refers to the awareness of how the body is moving in space and receives sensory inputs from the receptors within muscles, tendons, and joints [27]. Proprioception on the other hand, cares only about where the body is in space, regardless of how it got into that position. When creating an analogy for robots, the cutaneous sense corresponds to the sense of touch provided by dedicated sensors of touch [11, 12, 27, 52, 126] and the kinesthetic sense of touch is the one derived from a tension or stress from actuation, which is the focus of this dissertation.

In this dissertation, we are motivated by the problem of designing control architectures that enable the sense of touch for compliant soft robotic fingers with sensing capabilities. There are challenges in producing the desired controllable behavior from a soft robot due to the soft nonlinear materials. The softer and more compliant the robot becomes the more it's unconstrained, making its movement unpredictable and difficult to control. The decreased predictable movement makes the robot's configuration at a given time unknown, especially when it is interacting with objects or the environment. However, by integrating a sensory system, we can make observations from open-loop experiments to infer properties of the finger's motion and how it interacts with its environment. We make use of the benefit that the soft material dissipates energy through deformation when coming into contact with objects, and use the embedded sensors to determine the hand's configuration for defining whether the finger has made contact with an object. We consider speed and precision as having less importance than safety and reliability when performing tasks within unstructured environments inhabited by humans.

Our method focuses on designing a framework for a kinesthetic touch approach based on inferring models for a finger-like shaped soft robotic actuator innervated with sensors. We

design a reference tracking controller, which uses the finger's curvature measurements and enables the sense of touch. The tracking control allows the finger to follow a desired curvature reference signal, which results in a controllable motion of the finger over configurations in the range between full flexion and extension of the finger or maintaining a constant curvature. The tracking error of the control loop is negligible when the finger's motion is unimpeded, i.e., not in touch with any object. When the finger makes contact with an object, the error increases and its intensity acts as the amount of stress on the finger. The proposed method for identifying the model of the finger curvature addresses environmental variations as well as variations in fabrication due to material or human factors that can have an affect on the finger dynamics. We explore a variety of tasks where the sense of touch approach allows for the robot's workspace to be visualized. If the finger is dragged, the curvature is reduced as it comes into contact with an object and returns to the reference curvature once it is no longer in contact. The tracking error can be used to detect objects, their edges, and the level of the error can be used as an indication of change in object heights. Controlled motion of the finger also allows for tapping objects and visualizing objects. Moreover, this method of touch detection can be used for other soft robotic fingers with curvature sensors [55, 89, 128, 129]. All these capabilities of a single finger can be further exploited on a multi-finger robotic gripper and used as a building block for its controlled grasp. Our work is a significant step towards control principles and mode of uses of such soft robotic grippers in contact focused tasks. To the best of our knowledge, information about a finger-like actuator position and an effort to control it as measured by a tracking error of feedback control loop for the curvature of the finger has not been previously considered to detect and maintain a contact between the actuator and an object in its operating space.

1.2 Thesis Contributions

The research presented in this dissertation is applied in nature and pursues four main contributions. The primary problem studied revolves around feedback control approaches for the sense of touch for compliant finger-like actuators. We aim to develop new frameworks that apply existing methods for modeling and feedback control of both static and dynamic models describing the finger curvature. The contributions of the research are the following:

1. utilize the properties of how a somatosensitive capable soft finger-like actuator interacts with their environment to develop a control architecture that motivates the finger and is capable of detecting and maintaining contact with an object

2. rethink how force contact sensing can be realized without the use of a force sensor to develop a kinesthetic feedback approach for contact detection and validate the control architecture on a newly designed low-cost soft finger for a static model

3. improve the kinesthetic feedback controller by identifying the dynamics of the finger curvature sensor

4. demonstrate the kinesthetic feedback approach for contact-rich tasks that visualize the robot's workspace

There is a rich body of work exploring different methods of modeling and control for soft robotic components. However, many of these works focus on the challenges that the soft body presents such as how its compliant nature can make it difficult to predict positioning and interactions with external objects, and some are attempting to apply traditional rigid-link

manipulator techniques. For contribution (1), we choose instead to observe how the soft actuator operates in both a free and blocked displacement, where we refer to free displacement as an operating space with no external objects in the actuators path, and a blocked displacement as there being an object in the actuators path that interferes with it's motion. In this work, we focus on soft robotic actuators that are embedded with at least a proprioceptive curvature sensor along the dorsal ridge of the finger and may also contain a tactile force sensor on the underside of the finger. By observing how the sensory system of the actuator responds in these two scenarios we can design a control architecture that benefits from the compliant nature of the actuator.

We analyze previously published data collected on both the curvature and force sensor of a soft somatosensitive actuator [118] for free and block displacements to define static characteristic relations between the change in resistance of the sensors with respect to inflation. We show that both relations are monotonically increasing and thus can be used to inform simple linear controllers. We utilize the prior knowledge and characteristic equations that quantify the resulting change in resistance on the curvature sensor when the finger makes contact with external bodies to design a control architecture. The constrained bending is used in a threshold based switching rule as a way to determine if it is in contact with an object, and signals the control architecture to maintain contact on the object by switching to force control designed around the force sensor. The resulting control architecture provides a firm touch of a soft robotic finger with an object, which we see as a prerequisite for the robust grasping of an object with a soft robotic multi-finger gripper.

Force control is a challenging and open problem. The previous approach used the force sensor to maintain contact with an object. However, force sensor can be unreliable due

not only to being noisy but also because of the dependence on the physical location of where contact is made with the object. This location will determine whether enough contact is made between the object and the sensor to activate the sensor. Our second contribution addresses this issue and we rethink our control approach and design a control architecture that only relies on the curvature sensor as feedback. To achieve a control architecture that does not require the force sensor we consider different reference inputs for a reference tracking curvature controller. To set the finger in motion we use references that simulate a periodic flexion and extension motion such as a triangular or sinusoidal wave. Once contact is detected, we maintain contact by holding a constant curvature reference. We continue to employ the threshold based switching rule for contact detection but now we command the controller to switch between reference signals instead of controllers for different sensors. We refer to this as a kinesthetic feedback approach to touch inspired by the human kinesthetic sense modality. The approach is validated for a new soft pneumatic actuator with embedded intrinsically soft fluidic sensors that is designed to minimize the need for specialized equipment. While this approach has limitations, data recorded during the experiments of the system presented further insight into the dynamics of the soft finger and actuating system.

Using the static characteristic relation approach, we are required to limit how quickly the finger can move since we have not designed the controller to account for the dynamics of the finger. In many scenarios it is desirable for a robotic system to have the ability to move quickly. For contribution (3) we apply a system identification method to identify the dynamics of the finger curvature. System identification approaches to modeling and control are well suited for soft robots. Due to their compliant nature, soft robots pose less of a physical threat to themselves

or their surroundings when subjected to random control inputs than conventional rigid-bodied robots. This makes it possible to safely automate collecting input-output data over a range of operating conditions as long as the inputs remain within the soft robots own constraints, i.e. not using a pressure that will cause a bladder chamber to rupture. We examine two approaches for determining the best order of the model for two different compliant fingers. The first approach applied to a simulated compliant multi-link finger with flexible joints assumes nothing about the model and identifies not only the parameters, but also the order of the model. To check that the identified model does not overfit to noise, we apply a model reduction technique to look for pole-zero pairs that could be canceled and still provide a good approximation of the originally identified model. To decide if removing a pole-zero pair still provides a good approximation we analyze the model approximations in the frequency domain along with the original model. The second approach applied to the same physical soft actuator developed from contribution (2) benefits from the compliant nature of the soft actuator. To determine the model order we first apply informative open-loop experiments where the finger is tapped to visualize the transient response to an impulse. From this experiment we were able to select an order for the model and identify the unknown parameters.

Lastly, this dissertation demonstrates the effectiveness of the kinesthetic feedback approach through contact-rich tasks that visualize the robot's workspace. We incorporated the finger system onto a Stretch RE1 mobile robot and planned trajectories for the robot with the soft finger as the end-effector. The tasks include keeping the finger in a firm touch with an object, detecting the object edges and visualizing the operating space based on the sense of touch. To perform these tasks, we collect data about the tracking error for the finger that is mounted on a

robot arm, and use these tracking error data along with varying references to motivate how the finger interacts with the environment. These contact based tasks highlight that the error signal, which acts as a measure of the finger stress resulting from contact with an object, contains robust information regarding the finger's sense of touch and how it interacts with the environment.

In this work, we infer models from observations and utilize the properties of how soft finger-like actuators interact with their environment. We focus on creating a closed system that does not rely on any external sensing such as motion capture systems for determining positioning of the actuator or contact with objects. By not relying on external sensing our approaches are more applicable to real-world unstructured environments. We show that while soft robots have less predictable movement due to their compliant nature that the compliance overall is a benefit that allows for simpler control methods. The methods focus heavily on the type of actuator presented, and admittedly, many applications in soft robotic systems would be more appropriately tackled by other means of modeling and control. However, it is our hope that this dissertation will provide evidence for some of the benefits to the simple linear feedback control approach as applied to nonlinear, soft robotic systems with embedded sensing capabilities. Moreover, it is intended to help illuminate the role of kinesthetic sensing in designing feedback control policies for soft robots.

1.3 Outline

This dissertation is organized as follows. Chapter 2 summarizes the state of available literature. Chapter 3 introduces some of the preliminary modeling and mathematical concepts

and tools used in this dissertation. In doing so, we also present a road map of the remainder of the dissertation and further motivate the inclusion of the results in the sequel. Chapters 4-7 consist of appended papers detailing research results. Brief summaries of each of these papers appear below. Chapter 8 concludes the dissertation with a summary of discussion and directions for future research.

- **Chapter 4: Movement Error Based Control for a Firm Touch of a Soft Somatosensitive Actuator**

  In this paper, we propose a control architecture with the aim of providing a firm touch of a soft somatosensitive actuator (SSA) finger with an object. The two main components of the architecture are a reference tracking curvature controller and a force controller. The first one sets the finger in motion which is blocked if the finger comes into contact with an object. The result of such an event is that the finger bending is constrained, and the tracking error of the curvature controller increases. Once the error exceeds a predetermined threshold value, there is a switch from the reference tracking to the force controller, which maintains the finger in contact with the object. Therefore, the proposed architecture accounts not only for sensory data, but also for the error of the movement towards the touch. The proposed control architecture is illustrated with numerical simulations.

- **Chapter 5: Controlled Multilayered Soft Actuator Containing Embedded Intrinsically Soft Sensors**

  With this paper, we rethink and validate the role that feedback control can play in the control of soft finger-like actuators. In our method of touch detection, we use feedback

control to create a reference tracking for a curvature sensor resistance. The presented method uses the disagreement between the controlled curvature sensor resistance and its reference signal to detect the contact between the soft finger and an object. Once the contact is detected, the soft finger stops moving and keeps in contact with the object, which is a prerequisite for the use of soft fingers for grippers with controlled grasping. Our method of touch detection does not require a dedicated force sensor at the distal end of the actuator, therefore, it overcomes the necessity for the co-location between the point of contact and the corresponding sensor. To fully validate our method, we present the manufacturing method and components of our test system showing that the system is built without exceptional requirements. The results are illustrated with experiments in which the finger uses only its proprioceptive curvature sensor resistance to detect contact with an object and remain in contact with it.

- **Chapter 6: Compliant Proprioceptive Touch without a Force Sensor: A Kinesthetic Feedback Control Approach**

  The focus of this paper is on an approach that provides a kinesthetic sense of touch for compliant soft robotic fingers. The approach is based on a reference tracking feedback control for the finger curvature. The approach uses the reference tracking error as a measure of the finger stress resulting from a contact with an object. The study is based on a computer simulation model that mimics essential properties of a real soft finger. To achieve the control, we identify the finger model. The model combined with the reference signal informs the controller order and parameters. Then, we collect data about the tracking

error for the finger that is mounted on a robot arm. We use these tracking error data and references to define tasks requiring the sense of touch. The tasks include keeping the finger in a firm touch with an object, detecting the object edges and visualizing the operating space based on the sense of touch. All of these tasks are successfully demonstrated in the result section of the paper.

- **Chapter 7: Proprioceptive Modeling and Control of a Soft Somatosensitive Actuator for a Kinesthetic Feedback Approach to Touch**

  This paper presents a framework for a kinesthetic touch approach based on system identification for a finger-like shaped soft robotic actuator innervated with soft fluidic sensors. We design a curvature sensor signal reference tracking controller utilizing the embedded curvature sensor for proprioceptive sensing. The approach uses the reference tracking error as a measure of the finger stress resulting from contact with an object. We collect data about the tracking error for the finger that is mounted on a robot arm, and use these tracking error data and references to accomplish tasks requiring the sense of touch. The tasks include keeping the finger in a compliant touch with an object, detecting the object edges and visualizing the operating space based on the sense of touch. These tasks demonstrate that the error signal contains robust information regarding the finger's kinesthetic sense of touch and how it interacts with the environment.

Chapter 2

Related Works

In this chapter, we identify and describe some previous research efforts on topics that are relevant to this dissertation. A key challenge in producing desired behavior from a soft robotic component is to deal with the complexity of its motion. We begin with a review of some results in the field of soft robotics, modeling and control, and sensing capabilities, in particular, we focus on results related to manipulators and finger-like actuators. This includes a review of analytical kinematics and dynamic models, and data-driven approaches to modeling; control methods; and sensor technologies that have been used with soft actuators to improve control and modeling. Many of these topics are coupled together, therefore much of the literature touches on two or more of these areas.

2.1 Modeling and Control

Due to the variety of soft robotic systems, a standard approach to modeling soft robots does not exist yet. The literature presents a variety of approaches from static kinematic

to dynamic models as well as data-driven approaches for control [42, 105]. Many of these approaches are dependent on the type of soft robot system as well as its application. We identify and describe selected analytical kinematic and dynamic modeling approaches, respectively, and present some recent data-driven approaches to modeling. Analytical models are based on material properties and classical mechanics that define the system's behavior prior to putting the system to use. Data-driven approaches on the other hand require inferring models from observations of data collected during experiments with the physical system.

2.1.1 Kinematics Modeling and control

One of the most common kinematic models presented in the literature is the constant curvature (CC) model that provides a description of the shape of continuum robots under the assumption that bending occurs of constant curvature [124]. This model has been found to be a good approximation if the manipulator is uniform in shape and symmetric in actuation design, external loading effects are negligible, and there are little to no torsional effects. The CC model is often used to form a piece-wise constant curvature (PCC) model for multi-segment manipulators where the assumption that each section of bending occurs of a constant curvature. The PCC model has been used for a multi-segment planar manipulator for closed-loop curvature control using an external camera tracking system for visual feedback about the position [32, 61, 81, 83]. The design uses a two-staged cascaded PI/PID control approach to adjust the drive cylinders affecting the curvature of individual segments for the planar multi-segment arm. A single finger gripper is add onto the same manipulator design and uses an open-loop controlled bending motion for grasping [62]. The same model and low-level control was also applied to a multi-segment

manipulator that moves in three dimensional space [84]. The PCC model is limited to soft robots where each body segment has to be assumed to deform with constant curvature. In the case of finger-like actuators with a single soft segment, linear PID controllers have been demonstrated to achieve position control [45, 89, 128].

While analytical modeling of soft robots is challenging due to the soft materials, control designs for soft robots often benefit from that same softness. Due to the compliance of the materials, soft manipulators and grippers more robustly grasp objects and navigate environments of unknown shapes. Some researchers have even shown how entirely soft and compliant robots can use collisions to assist in accomplishing the task [28, 81, 85]. The compliance of the hand in [28] enables it to shape to the surfaces of an object in response to contact forces through simple open-loop control.

A benefit to using a model is that filters like extended Kalman filters can be used to produce estimates of unknown or noisy variables for more accurate control. A pseudo-rigid quasi-static model is used with optical fiber for curvature sensing for an adaptive extended Kalman filter approach [77]. Modeling uncertainties often still arise due to the simplifying assumption, and unmodeled dynamics. Although, often times these uncertainties can be addressed by the compliance of the design.

2.1.2 Dynamic Modeling and control

It is often desirable to move robots quickly in which case a dynamic model is needed. A Lagrangian approach is often used to describe dynamic models for both rigid-bodied and soft robots. However, there is a difference in the way energy is stored in soft versus rigid

manipulators. In a soft pneumatic or fluidic actuators the input pressure energy is stored as both strain and gravitational potential energy. Both of these forms of potential energy can deform the manipulator and transfer to kinetic energy. However, there are losses in transmission lines and system. In [85] they present how to find the different components of energy for a single arm segment and then use a Lagrangian approach to derive the equations of motion with respect to the segment's generalized coordinate. [85] provides one of the first dynamic motion control for soft manipulator using iterative learning control to plan and execute a grabbing maneuver. The grabbing is defined as bringing the arm's end effector to a user specified, statically unreachable goal point with near zero tip velocity. A feedback control architectures that is able to compensate for dynamical forces, and stabilize a desired trajectory in the curvature space is presented in [61] using a motion capturing system. A PD controller is used with dynamics compensation in a hierarchical controller that uses optimal control for a planning algorithm to control the tip position. The dynamic model of a soft robot has been shown to be reduced to first-order dynamical model approximation due to the high damping and low inertial properties [44] for developing closed-loop task-space dynamic controllers.

2.1.3 Data-driven Modeling and control

The nonlinear properties of soft materials and high dimensionality of soft robots still presents difficulties for these analytical models. Even with all of the proper modeling and defining parameters, there will still be some uncertainties. Data-driven system identification approaches can help to identify unknown parameters in models as shown in [34, 85, 116]. Another approach to coping with it is to use machine learning techniques, like neural networks

[20, 114], reinforcement learning [113], or nonlinear auto-regressive with exogenous inputs models (NLARX) [44, 58, 74, 75]. A feedforward neural network is used to learn the reduced first-order dynamical model approximation presented in [44] for soft robots by simplifying the planning and sensory feedback process with minimal effects on the controller accuracy. The Koopman operator theory has been used as a way to construct a linear dynamical model [17] and combined with model predictive control [16]. Similar to our approach, this method gives an explicit control-oriented linear model rather than a black-box input-output mapping such as that given by neural networks.

These methods are useful for handling uncertainties during fabrication where each robotic component could have slightly different results do to human factor. Some designs are also sensitive to environmental impacts like humidity and temperature. Elastomers are also susceptible to wear and fatigue, especially under excessive stretching. One way to reduce the amount of wear is to avoid regularly inflating at maximal bounds, but high pressure ranges are often necessary to apply enough grip strength. Some work has shown that adding a reinforcement helix can help limit the stretching [28]. System identification methods can be reapplied over time to identify models periodically so that the stretching can be taken into account.

2.2 Sensor Feedback Control for Soft Finger-like Actuators

Soft compliant robot components have less constrained movements which can be unpredictable. This unpredictability can result in unexpected configurations at a given time, especially when the component is in contact with objects in the operating environment making

them more difficult to control [105]. The mechanical finger design and embedded sensors on soft robotic components can greatly enhance the ways in which finger-like actuators interact with objects [110] and how they can be controlled. Many closed-loop controllers for soft robotic systems rely on external motion capture camera systems for feedback on position or curvature control [62, 83, 85]. However, in order for soft robots to be more applicable in unstructured environments they need to be able to sense their configurations and external forces without the use of the external camera systems.

To improve a soft robot's bodily awareness and how it interacts with its environment, researchers have embedded commercial sensors like bend and force sensors [38, 45, 56, 74]. Other sensors that have been used for determining curvature are optical [129], magnetic [95], and strain sensors [118]. Many of the mentioned options show promising results. However to fully capture the soft bodies sensor technologies that are intrinsically soft and stretch and compress when the robot is deformed have been developed. These are composed of ionically conductive fluids used to create soft sensing skins [23, 93] or are embedded directly during the manufacturing process [118]. The optical waveguides [129] have demonstrated capabilities in both proprioception and external forces at the fingertip. The anthropomorphic hand demonstrated detecting shape, texture, and softness in objects, such as selecting a ripe tomato out of other non-ripe tomatoes. However, this was all done using only open-loop control to focus on demonstrating the sensing capabilities.

Many of these sensors are used for closing the loop or developing a model for the soft actuator using the sensor measurements [55, 56, 89]. [114] trained a long short-term memory (LSTM) network as the kinematic and force model to estimate the tip position and contact force

of a soft finger with embedded cPDMS sensors. Sensors can also be utilized for designing filters for state estimation [74, 75, 77] and observers that can improve the position without the dependence on camera systems. Measurements from an embedded flex sensor were used with an Extended Kalman Filter to estimate the curvature of a soft finger [74, 75].

A modular gripper formed from three soft fingers was outfitted with additional commercial resistive bend sensors to provide proprioceptive grasping [55] where the bend sensor was used to determine finger shape to identify grasped objects, and was further improved upon by incorporating external resistive force sensors on the fingertips for contact detection and force-controlled grasping [56]. However, the authors found that the force sensor was not always reliable for determining contact if the object was too small or the fingers did not make contact in a location that activated the sensor. Our approach presented in Chapters 5-7 alleviates this issue.

Finger-like actuators are usually equipped with sensors closely following the current shape of the actuator and measuring its curvature [114, 127]. While much of the literature focuses on curvature sensing, the cutaneous sense that corresponds to the sense of touch can also be provided by dedicated sensors [11, 12, 27, 52, 126] along with force sensors such as [119] which is attached to the tip of the finger-like actuator or the one which is embedded in the actuator's body [56, 117, 118]. The soft finger presented in this work has embedded resistance based curvature and force sensors. In our design, both sensors are soft and capable of stretching with the actuator's body. While the design has a force sensor, we focus on using the curvature sensor for detecting touch. Using only the curvature sensor along the dorsal ridge of the finger allows for the collocation of contact between the finger and object to be of less importance.

In order to perform tasks requiring contact there has been many research efforts in

force control aimed at traditional rigid robot manipulators for contact dynamic modeling using various methods such as operational space control (OSC) [63], impedance control [53] and the hybrid position/force controller [101] among others [47]. When working with rigid robots, instabilities or bouncing of the manipulator against a stiff environment can result during the non contact to contact detection. To address the bouncing effects against the environment a hybrid control strategy for a switched position–force controller with hysteresis is considered, where the controller switches from motion to force control when contact with the environment is made relying only on contact force measurements [19]. Some of these methods have inspired efforts in force control of soft robotic manipulators to achieve surface following [31]. While this work shows promising results, the method is dependent on an external motion camera system for detecting contact. Another common method for force control seen in the literature is based on using a dedicated contact sensor. However, many existing robotic fingers with contact sensing are sensitive to precise location of the object below the contact sensor to be triggered indicating that contact has been made reducing the effectiveness in real world scenarios. This exact issue was acknowledged in [56].

With our method, we re-think contact sensing for soft robots and present a novel approach to contact detection. We use the very compliance which makes soft robots difficult to control as a source of state information, using the compliance to our advantage. We focus on a different sensing modality centered around proprioception. Our control architecture is a two state contact detection strategy that starts with a freely moving finger not in contact, detects when contact is made, and switches between the non-contact motion to maintaining contact. The finger is first set in motion assuming it is not already in contact. The pneumatic inflation of the

finger continues, increasing contact force between the finger and the object, deviating the pose of the finger from its free-displacement state, which is fundamentally different from traditional rigid robots (where increased torque would cause minimal deflection of the rigid actuator and would likely damage the object or motor). Once the touch with an object is detected, the finger stops moving and keeps contact with an object. We denote this state of the finger a *compliant touch with an object* and achieving this state is a prerequisite for controlled grasping with a soft robot gripper. Note that due to the compliant nature of soft fingers, it is less crucial for the generalized grasping strategy [55] to limit contact force, as the gripper will simply deflect rather than crushing the object. The proposed detection method overcomes the necessity for the co-location between the point of touch and the corresponding sensor. Moreover, while we have presented our method on a specific finger-actuator, our presented method can be used on a wide variety of soft robotic fingers with curvature sensors [55, 89, 128, 129]. It is not necessary for the sensors to be composed of an iono-gel material, but important characteristics of these curvature sensors being that they have a monotonic relation as bending increases and a detectable decrease in sensing when becoming impeded compared to moving in free displacement. The specific values presented in this work for our models and control parameters apply to the actuator developed for validation purposes of our methods, but the overall framework presented can be generalized to the broader community using soft fingers with curvature sensing abilities.

To the best of our knowledge, this dissertation and the papers within are the first attempt at designing a kinesthetic feedback approach for the controlled sense of touch with out a force sensor for soft finger-like actuators. Our approach utilizes both a static resistance-pressure model and an identified dynamical model of the finger curvature. The method uses the resulting

error of the curvature reference tracking controller in such a way to anticipate contact with an object for visualizing the robot's workspace.

Chapter 3

Preliminaries and Road Map

This chapter serves two purposes. First, it introduces some of the preliminary concepts and mathematical tools that will be used in later sections. This chapter also assists in drawing a road map of the remainder of this dissertation and motivating the inclusion of the results of Chapters 4-7. This work relies on the idea of inferring models from observations. Section 3.1 presents standard results from the use of Lyapunov functions to guarantee stability of the closed-loop system toward some goal. In Section 3.2, we provide brief comments on the formulation of the feedback control architectures that rely on the finger's embedded sensory system and measure of the quality of the finger's motion necessary to achieve a compliant contact, leading to a formulation of a threshold based switching rule for the feedback control problem. Section 3.3 introduces the linear least squares method for finding an Auto-Regressive with exogenous inputs (ARX) model that gives an explicit control-oriented linear dynamical model.

3.1 Lyapunov Stability for Static Characteristic Relations

We first introduce the methods applied for determining the models used in this dissertation to describe the finger motion through a static characteristic relation of the curvature relative to inflation pressure, although details specific to Chapters 4 and 5 can be found there. When inflated, the finger bends freely in an arc. The curvature and contact sensors directly embedded within the actuator body are designed to deform when stretched or compressed, which consequently changes the sensors' resistances when the finger inflates, bends, and makes contact with external bodies. Data was collected from experiments in which the finger was inflated with a sequence of pressure values and measured the resistance of the curvature and the contact sensors. The collected data is used to characterize the change in resistance, ΔR, given by $\Delta R = R - R_0$, where R_0 is the initial resistance, for the embedded curvature and contact sensors.

The finger was positioned in both free displacement where there are no objects in its path, and blocked displacement where the motion of the finger becomes obstructed by an object. We can find a fit to the data to obtain an approximate relation of the inflation pressure against the change in resistance of the sensor during the free and blocked displacements. From the data and relations, we observe and conclude that all relations between the inflation pressure and sensor measurements vary with the pressure and they monotonously increase.

Let us assume that we want to design a single feedback control loop to achieve a certain configuration, i.e., the position of the free moving finger-like actuator. Under the assumption that we expect to move the finger slowly, we can realize that this control is equivalent to the closed loop control of the change in the curvature sensor, ΔR_{cf}. Therefore, the control that keeps ΔR_{cf}

at a reference value $\Delta R_{\text{cf}}^{ref}$ can be designed using the Lyapunov function

$$V = \frac{1}{2}(\Delta R_{\text{cf}}(p) - \Delta R_{\text{cf}}^{ref})^2 \tag{3.1}$$

where the control variable is the inflation pressure p. Based on the nonlinear control theory, $\Delta R_{\text{cf}} \to \Delta R_{\text{cf}}^{ref}$ if we can find such a control p, so that the time derivative $\dot{V} < 0$ for all $\Delta R_{\text{cf}} \neq \Delta R_{\text{cf}}^{ref}$. The derivative is

$$\dot{V} = (\Delta R_{\text{cf}}(p) - \Delta R_{\text{cf}}^{ref}) \frac{\partial(\Delta R_{\text{cf}}(p))}{\partial p} \dot{p} \tag{3.2}$$

and we are free to choose the form of $\dot{p}$ to obtain $\dot{V} < 0$, so we choose

$$\dot{p} = -K(\Delta R_{\text{cf}}(p) - \Delta R_{\text{cf}}^{ref}),\ K > 0 \tag{3.3}$$

where K is a tuning parameter of the controller. The substitution of $\dot{p}$ in (3.2) yields

$$\dot{V} = -K \frac{\partial(\Delta R_{\text{cf}}(p))}{\partial p} (\Delta R_{\text{cf}}(p) - \Delta R_{\text{cf}}^{ref})^2 \tag{3.4}$$

Under the selected control $\dot{p}$, it guarantees $\dot{V} < 0$ for $\Delta R_{\text{cf}} \neq \Delta R_{\text{cf}}^{ref}$ as long as $\frac{\partial(\Delta R_{\text{cf}}(p))}{\partial p} > 0$, which is correct, for a control input, $p > 0$ (kPa) since the resistance-pressure relation is monotonically increasing. Consequently, the control law (3.3) guarantees $\Delta R_{\text{cf}} \to \Delta R_{\text{cf}}^{ref}$ and can be rewritten as

$$p = -K \int (\Delta R_{\text{cf}}(p) - \Delta R_{\text{cf}}^{ref}) dt. \tag{3.5}$$

This is an integral control, i.e., I-only controller with the parameter $K > 0$ which can be tuned to achieve a desired performance in keeping $\Delta R_{\text{cf}} = \Delta R_{\text{cf}}^{ref}$. However, from linear system control theory, we know that in our system such a controller cannot guarantee that the error of tracking references goes to zero for reference signals that are different from the step function. A periodic reference signal sets the freely moving finger in a motion that resembles the motion of a finger

that samples points in space to touch. Similar types of motion are also an integral part of the opening and closing of multi-finger grippers. We consider here the controller for tracking references that are composed of ramp signals. To account for that and achieve the convergence of the tracking error to zero, we consider second order linear controllers composed of double integrator. The control parameters, gain and zero(s), are tuned using the discrete root locus analysis with criteria of the damping ratio of $\zeta = 0.707$ on the root locus diagram while also trying to optimize rise and settling times in the step response. Our design choices were chosen to try and have a responsive controller without too much oscillations or overshoot. Through the root locus design process we ensure the controller meets our specified requirements for the desired system performance in tracking the ramp signal. Specific parameter values are presented in Chapters 4-7.

The controller presented in the following chapters is a curvature sensor reference tracking controller. The work of this controller results in the curvature sensor change of resistance ΔR_{c} values to follow the reference $\Delta R_{\mathrm{c}}^{Ref}$ signal. This is equivalent to the resistance R_{c} values to follow R_{c}^{Ref} reference signal since $\Delta R_{\mathrm{c}} = R_{\mathrm{c}} - R_{\mathrm{c}}(0)$. For a correctly designed reference tracking controller and freely moving finger, we expect a small tracking error $R_{\mathrm{c}}^{Ref} - R_{\mathrm{c}}$. Therefore, the tracking error signal can be used to detect contact of the finger with an object in the operating environment [13].

3.2 Threshold Based Switching Rule

In Chapter 4, the control architecture contains two controllers informed from the above analysis, (1) a closed-loop reference tracking curvature controller that utilizes the curvature sensor as feedback and (2) a force controller that uses the force sensor feedback to close the loop. The reference tracking controller sets the finger in motion, and the error signal from this controller is used to determine when to switch between the two controllers. The control architecture was motivated by the embedded sensory system of the Harvard SSA, and utilizes resistive curvature and contact sensor measurements based on previously published data [118]. However, in reality it is common that sensed forces are noisy and dependent on contact location. This issue is tackled in Chapter 5 where we instead use only the reference tracking curvature controller and instead switch between reference inputs to maintain contact. This method allows us to avoid using the force sensor for determining if contact was made which removes the vulnerability of contact location with the object.

3.3 Auto-Regressive with Exogenous Inputs Model via the Linear Least Squares Method

We next introduce the methods applied for determining the models used in this dissertation to describe the finger motion through a dynamical model of the curvature sensor, parameter details specific to Chapters 6 and 7 can be found there. For the model of the finger curvature

dynamics, we use the linear ARX discrete time model [73]

$$y(z) = G(z)p(z) + H(z)e(z) = \frac{B(z)}{A(z)}p(z) + \frac{1}{A(z)}e(z), \tag{3.6}$$

where $p(z)$ is the z-transform image of the discrete time control variable $p(t_k)$, $y(z)$ is the z-transform image of the curvature measurement variable $y(t_k) = \Delta R_{cf}(t_k)$, and $e(z)$ is the z-transform of the model error signal $e(t_k)$, and $k = 1, 2, ...$ The model sample time, $t_{k+1} - t_k$, is determined for each system based on system capabilities, and $B(z)$ and $A(z)$ are polynomials of the transfer function $G(z)$ defined as

$$A(z) = 1 + a_1 z^{-1} + \cdots + a_n z^{-n} \tag{3.7}$$

$$B(z) = b_1 z^{-1} + \cdots + b_m z^{-m}, \tag{3.8}$$

where $n \geq m \geq 1$. In the time domain, this model results in the linear difference equation

$$y(t_k) + a_1 y(t_{k-1}) + \cdots + a_n y(t_{k-n}) = b_1 p(t_{k-1}) + \ldots$$
$$\cdots + b_{m+1} p(t_{k-m+1}) + b_m p(t_{k-m}) + e(t_k), \tag{3.9}$$

which describes the relationship between the input $p(t_k)$, output $y(t_k)$ and the model error $e(t_k)$, $k = 1, 2, ...$ For this model, we estimate the vector of unknown coefficients

$$\theta = \begin{bmatrix} a_1 & \ldots & a_n & b_1 & \ldots & b_m \end{bmatrix}^T, \tag{3.10}$$

as well as the numbers n and m, i.e., orders of the transfer function polynomials $A(z)$ and $B(z)$. We estimate them based on the assumption that the sequence of model errors $e(t_k)$ is a white noise zero-mean.

Procedure: We identify the finger dynamics from an open-loop experiment, in which the input $p(t_k)$ is a biased pseudo random binary sequence within the pressure range defined as

$$p(t_k) = p_{bias} + p_b(t_k) \tag{3.11}$$

The bias is added to shift the range of the pseudo random input $p(t_k)$ to be only positive since the finger does not bend backward without an external force.

$$p_b(t_k) = \begin{cases} -p_b(t_{k-1}) & \text{with probability } \pi = 0.5 \\ p_b(t_{k-1}) & \text{with probability } \pi = 1-\pi \end{cases}$$

where $p_b(0) = \{-p_b, p_b\}$ should be an integer number within the range of acceptable inputs after the bias is applied, and randomly chosen to start the sequence. As an example,

$$\begin{aligned} p_b(t_k) &= \begin{cases} -1 & \text{with probability } \pi = 0.5 \\ 1 & \text{with probability } \pi = 1-\pi \end{cases} \\ p_b(0) &= \{-1, 1\} \end{aligned}$$

For the sequence $p(t_k)$, we collect the output data $y(t_k)$ and form the vector

$$\phi(t_k) = \begin{bmatrix} -y(t_{k-1}) & \dots & -y(t_{k-n}) & p(t_{k-1}) & \dots & p(t_{k-m}) \end{bmatrix}^T, \tag{3.12}$$

which can be used to express a one step prediction of the output $\hat{y}(t_k) = \phi(t_k)^T\theta$, for given θ. We can then estimate the parameter vector θ by minimizing the sum of squares of one step prediction errors for t_k, $k = n+1, ...N$ and $N >> n$ as

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N-n+1} \sum_{k=n+1}^{N} (y(t_k) - \phi^T(t_k)\theta))^2. \tag{3.13}$$

The minimization of the prediction errors results in the solution

$$\hat{\theta} = \left[\sum_{k=n+1}^{N} \phi(t_k)\phi^T(t_k) \right]^{-1} \sum_{k=n+1}^{N} \phi(t_k)y(t_k), \tag{3.14}$$

where the right side of the solution is completely defined based on input and output data recorded at t_k, $k = 1, ...N$.

Once we have recorded the data, we compute the estimate $\hat{\theta}$ for a nth order polynomial $A(z)$ $n = 1, 2, \ldots$ and using $m = n$ for the order of polynomial $B(z)$. To choose the order of the polynomial we explored two methods

1. increasing sequence of the polynomial $A(z)$ order $n = 1, 2, \ldots$

2. perform open-loop experiments to get a rough idea of the polynomial $A(z)$ order

In both cases, once n was defined we used $m = n$ for the order of polynomial $B(z)$. These two methods are described in section 3.3.1 and section 3.3.2. After determining the model, we compute the one step prediction for model to show that for the same input sequence, the model output closely follows the data from our experiments.

3.3.1 Model Reduction

We increase n until we do not see a significant change in the reduction of the sum of squares of prediction errors, which we use as a criterion to prevent the overfitting of the data. We obtain a model with the resulting parameters, $A(z)$ and $B(z)$ for the highest order model that we use to describe input-output relations from our data. To check if the model order can be reduced without affecting the model properties significantly, we look for poles that show up in pairs with zeros in their proximity. Pairs of poles and zeros of a transfer function that are in close proximity of each other can be canceled. Therefore, the presence of such pairs in our model indicates the possibility of overfitting, i.e., that our model is of a higher order than necessary, or unnecessarily

complex. To understand the impact of pole-zero pairs, we proceed with the analysis of the model in the frequency domain.

Model analysis: The analysis consists of a comparison of the original transfer function $G(z) = \frac{B(z)}{A(z)}$ with the simpler modified model $G'(z)$ obtained from $G(z)$ by removing one or more pole-zero pairs, while keeping the DC gain unchanged, i.e., $G'(1) = G(1)$. We compare the Bode plot of $G(z)$ and $G'(z)$, which is obtained by sequentially removing pole-zero pairs, looking for cases of $G'(z)$ that approximate $G(z)$ well both in magnitude and phase. We consider only the approximations that are well matched to the frequency characteristics of the original model, and out of those models choose the transfer function $G'(z)$ with the smallest order of polynomial. Not all pole-zero pair cancellations will result in a good approximation of the original higher-order model.

3.3.2 Informative Open-Loop Experiments

The compliant nature of soft robots means they pose less of a physical threat to themselves or their surroundings when subjected to random inputs than conventional rigid-bodied robots. This makes it possible to safely collect input-output data over a wide range of operating conditions. To have a better idea of model order for the soft finger, the non-inflated finger was tapped to see the transient response to an impulse. We were able to see from the data that ΔR_{cf} mimics the response of a damped harmonic oscillator. Therefore, we chose $n = 2$ and obtained a 2nd order model to describe the input-output relations from our data. We avoid using a higher order model than necessary which could result in overfitting to noise in the data.

Chapter 4

Movement Error Based Control for a Firm Touch of a Soft Somatosensitive Actuator

This chapter is a reprint of the paper

- M. Boivin, D. Milutinović, and M. Wehner, "Movement Error Based Control for a Firm Touch of a Soft Somatosensitive Actuator," in 2019 American Control Conference (ACC). IEEE, 2019, pp. 7-12.

4.1 Introduction

Soft robotic grippers made of soft materials with compliance that is comparable to the biological compliance of natural organisms [67, 105] are an attractive alternative to their rigid counterparts. The soft robotic grippers compliance not only simplifies tasks, such as grasping, but potentially reduces a risk of harm that could be inadvertently caused by traditional rigid robotic components and systems in applications where safety and adaptability to uncertainties in

human-centric operations are fundamental requirements [83, 105]. More detailed insights into the area of soft robotic systems, as well as challenges associated with creating robots of soft, nonlinear materials [79] are provided in the following references [67, 105, 110, 122].

A key challenge in producing desired behavior from a soft robotic component is to deal with the complexity of its motion. Motivated by the recent development of the finger-shaped Harvard soft somatosensitive actuator (SSA) [118], we propose a feedback control architecture that relies on the SSA's embedded sensory system and measure of the quality of the SSA's motion necessary to achieve a firm compliant contact, i.e., a touch with an object that is in its way, see Fig. 4.1. We consider this step as an important one in the development of reliable soft robotic grippers or hands assembled from multiple SSAs.

The softer and more compliant the robot is, its movements are less constrained, more unpredictable and more difficult to control [83]. The increased unpredictability of movements makes the robot configuration at a given time practically unknown, especially when it interacts with objects in the operating environment [55]. The inference of the hand configuration at any given moment [56] requires advanced internal sensing approaches called proprioception. Embedding stretchable, distributed proprioceptive sensors [24] in and exteroceptive sensors on soft grippers greatly enhances the ways in which the grippers interact with objects [110], and how the grippers can be controlled. Using these sensors to determine the gripper configuration can be useful to infer whether a grasp of an object is successful and robust, as well as the pose of the object [56].

Multiple soft robotic components and systems that already exist include soft fingers [62, 82, 118], grippers [55, 56, 80, 89], and multi-segment manipulators [81, 83, 85]. However,

only a subset of these have implemented closed-loop control. Some utilize closed-loop curvature control using external camera systems for visual feedback about the position [62,81,83,85], while others closed the loop using sensor measurements [55, 56, 89]. For example, a modular gripper formed from three soft fingers previously presented in [62, 82] was outfitted with additional resistive bend sensors to provide proprioceptive grasping [55], and was further improved upon by incorporating external resistive force sensors on the fingertips for contact detection and force-controlled grasping [56].

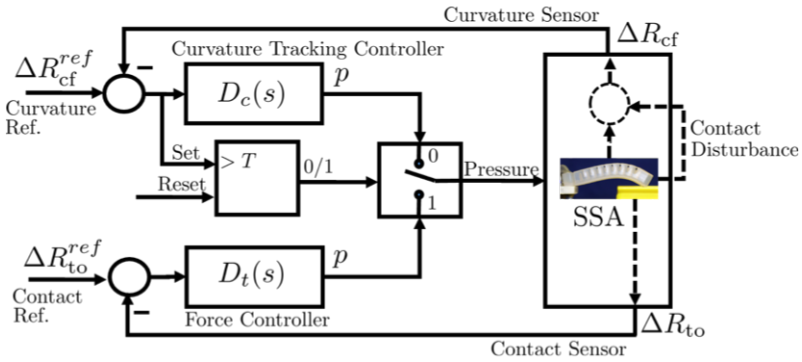

Figure 4.1: Feedback controller that tracks the curvature of the Harvard soft somatosensitive actuator (SSA) [118] and maintains contact with an object when present. The feedback is based on the curvature and contact sensor resistances. Once the curvature tracking controller error passes a threshold, the control switches from the tracking controller $D_c(s)$ to the force controller $D_t(s)$. Before the movement, the switch that dictates the currently active controller is reset to 0.

A firm touch of a soft robotic finger with an object is a prerequisite for the robust grasping of an object with a soft robotic multi-finger gripper, or an anthropomorphic soft robotic hand. To achieve this, we propose here a control architecture that is inspired by the approach to the control of kinematic problems in which the introduction of a stochastic prior is instrumental for the design of robust feedback controllers [3, 4, 21, 90] and the corresponding threshold based switching rule [86]. The control is deterministic and it employs the feedback control to set the

SSA in a certain type of reference movement, which serves as a *prior* for the movement, and uses the discrepancy between that movement and the actual one to robustly infer that the SSA is in contact with the object, so that the force control can be applied, see Fig. 4.1. Motivated by the embedded sensory system of the Harvard SSA [118], the control utilizes resistive curvature and contact sensor measurements. The presented control design uses previously published data on the Harvard SSA [118] and a deterministic model for the impact of the contact to curvature sensor measurements.

The paper is organized as follows. Section 4.2 introduces the SSA and the control problem. The SSA characteristics and preliminary control analysis are described in Section 4.3. The control architecture, reference tracking curvature and force controllers are described in Section 4.4. Section 4.5 presents numerical simulation results and Section 4.6 gives conclusions.

4.2 Problem formulation

The work in this paper is motivated by the finger-shaped Harvard SSA designed by the authors of [118]. The SSA is created via multimaterial, embedded 3D (EMB3D) printing and is innervated with a complex network of sensors. To create the desired bio-inspired sensing and actuation, the SSA contains multiple ionically conductive and fluidic features within elastomeric matrices. Proprioceptive and haptic feedback is demonstrated through the embedded curvature, contact and inflation sensors [118]. The SSA, resistive curvature and contact sensors are depicted in Fig. 4.2. These SSAs have also been created with temperature and deep-versus-fine touch contact sensing. Multiple SSAs can be assembled to form a soft robotic gripper.

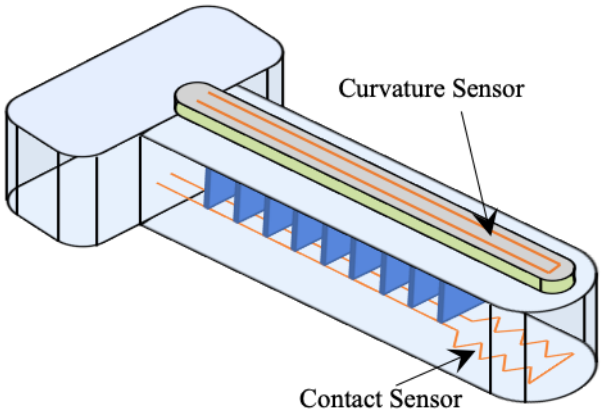

Figure 4.2: Schematic illustration of the Harvard SSA [118] showing the location of the curvature and contact sensors. A more detailed schematic can be found in [118].

When inflated, the SSA bends freely in an arc. The inflation, curvature and contact sensors directly embedded within the SSA's body are designed to deform when stretched or compressed, which consequently changes their resistances when the SSA inflates, bends, and makes contact with external bodies. Configurations of the freely moving deflated and inflated SSA can be seen in Fig. 4.3a-b. Starting from the deflated configuration in Fig. 4.3a, the increase of the inflation pressure bends the SSA in such a way that its tip moves clockwise along a circle-like trajectory. If the SSA is deflated from the configuration in Fig. 4.3b, the tip moves counter clockwise until the SSA is deflated and in the configuration shown in Fig. 4.3a. If the SSA's movement is blocked, the SSA's configurations for the same levels of the inflation pressures are significantly different, as shown in Fig. 4.3c-d in which the SSA's free movement is blocked due to the contact with an object.

Fig. 4.3 illustrates the flexible nature of the SSA's body and the impact of an object that is in the way of the inflating SSA to the SSA's configuration curvature. *Given the complexity of the SSA's body and its interaction with objects, the problem is to control the SSA to achieve its*

movement and a firm touch with an object that is in its way of motion during the inflation.

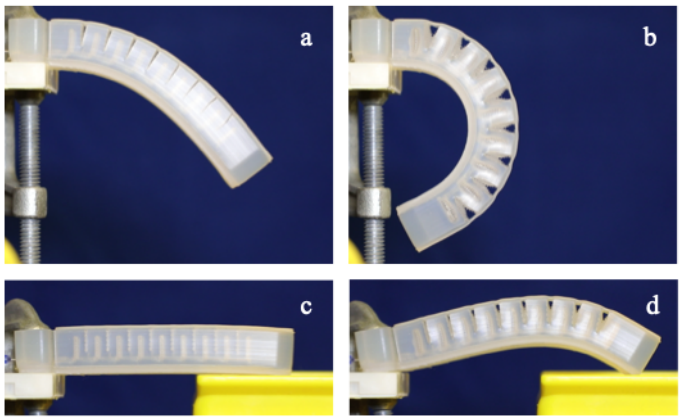

Figure 4.3: Images of the deflated and inflated Harvard SSA: a) Free SSA at 0 kPa; b) Free SSA at 165 kPa; c) Blocked SSA at 0 kPa; d) Blocked SSA at 165 kPa.

4.3 Data and Preliminary Control Analysis

Previously, the authors of [118] characterized the change in resistance, ΔR, given by $\Delta R = R - R_0$, where R_0 is the initial resistance, for the embedded curvature, inflation, and contact sensors during the free and blocked displacements. They collected data from the experiment in which the SSA was inflated with a sequence of pressure values and measured the resistance of the curvature and the contact sensors.

From the published data, we can plot a diagram of the inflation pressure against the curvature sensor resistance change (see Fig. 4.4) of the free moving SSA and fit the data to obtain an approximate analytical relation as

$$\Delta R_{\text{cf}}(p) = 0.0064p^2 + 0.93p + 1.1 \tag{4.1}$$

We repeat the same process for a diagram of the inflation pressure against the contact sensor

resistance change (see Fig. 4.5) of the SSA in contact with an object, similar to the one in Fig. 4.3c-d, and for which an approximate analytical relation is

$$\Delta R_{\text{to}}(p) = 0.4619e^{0.03763p} \tag{4.2}$$

Finally, we can also obtain from the data an approximate relation between the inflation pressure against the curvature sensor resistance change (see Fig. 4.6), in the case of the SSA in contact with the object, and the corresponding approximate analytical relation as

$$\Delta R_{\text{co}}(p) = 0.0055p^2 + 0.019p + 5.5 \tag{4.3}$$

In the original data, the authors of [118] also collected data from the contact sensor of a free moving SSA, but in that case, the SSA was not in contact with any object and the corresponding change of the resistance ΔR_{tf} was negligible, i.e., $\Delta R_{\text{tf}} \approx 0$. From the diagram and relations (4.1)-(4.3), we can conclude that all relations vary with the pressure and they monotonously increase. We illustrate the significance of it by the following preliminary analysis. Let us

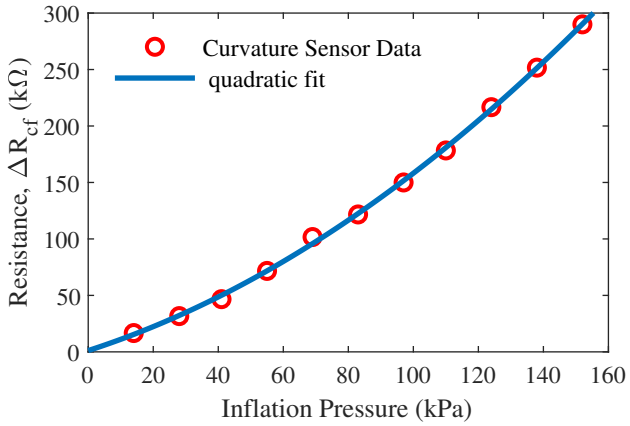

Figure 4.4: Quadratic function fit of the resistance change, ΔR_{cf}, for the curvature sensor as a function of inflation pressure during the free displacement.

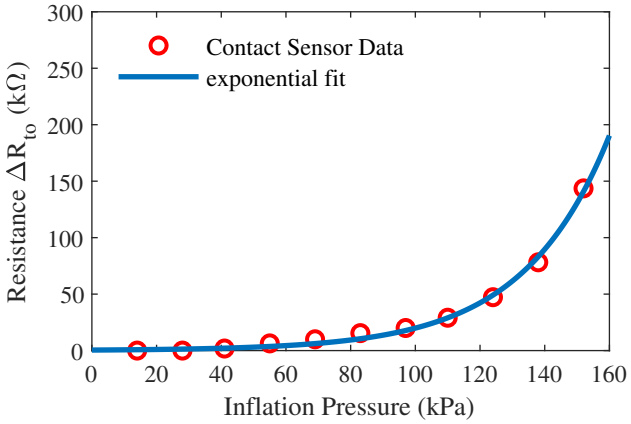

Figure 4.5: Exponential function fit of the resistance change, ΔR_{to}, for the contact sensor as a function of inflation pressure during the blocked displacement.

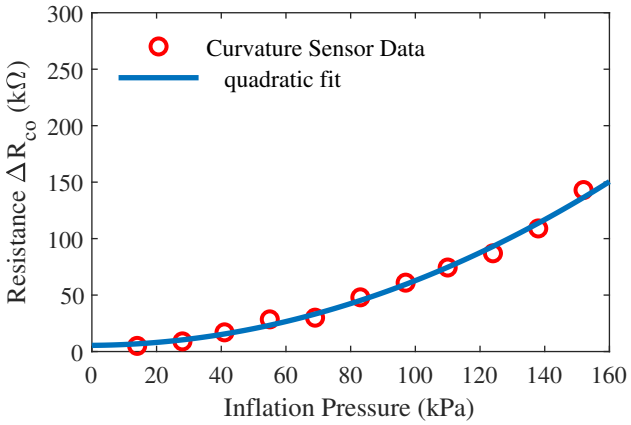

Figure 4.6: Quadratic function fit of the resistance change, ΔR_{co}, for the curvature sensor as a function of inflation pressure during the blocked displacement.

assume that we want to design a single feedback control loop to achieve a certain configuration, i.e., the position of the free moving SSA using the diagram in Fig. 4.4, i.e., the corresponding relation (4.1). Note that this approach is valid if we expect to move the SSA slowly and we can realize that this control is equivalent to the closed loop control of ΔR_{cf}. Therefore, the control that keeps ΔR_{cf} at a reference value $\Delta R_{\text{cf}}^{ref}$ can be designed using the Lyapunov function

$$V = \frac{1}{2}(\Delta R_{\text{cf}}(p) - \Delta R_{\text{cf}}^{ref})^2 \tag{4.4}$$

where the control variable is the inflation pressure p. Based on the nonlinear control theory, $\Delta R_{\text{cf}} \to \Delta R_{\text{cf}}^{ref}$ if we can find such a control p, so that the time derivative $\dot{V} < 0$ for all $\Delta R_{\text{cf}} \neq \Delta R_{\text{cf}}^{ref}$. The derivative is

$$\dot{V} = (\Delta R_{\text{cf}}(p) - \Delta R_{\text{cf}}^{ref}) \frac{\partial(\Delta R_{\text{cf}}(p))}{\partial p} \dot{p} \tag{4.5}$$

and we are free to choose the form of $\dot{p}$ to obtain $\dot{V} < 0$, so we choose

$$\dot{p} = -K(\Delta R_{\text{cf}}(p) - \Delta R_{\text{cf}}^{ref}), K > 0 \tag{4.6}$$

where K is a tuning parameter of the controller. The substitution of $\dot{p}$ in (4.5) yields

$$\dot{V} = -K \frac{\partial(\Delta R_{\text{cf}}(p))}{\partial p} (\Delta R_{\text{cf}}(p) - \Delta R_{\text{cf}}^{ref})^2 \tag{4.7}$$

Under the selected control $\dot{p}$, it guarantees $\dot{V} < 0$ for $\Delta R_{\text{cf}} \neq \Delta R_{\text{cf}}^{ref}$ as long as $\frac{\partial(\Delta R_{\text{cf}}(p))}{\partial p} > 0$, which is correct, since for $p \in [0, 160]$ (kPa), expression (4.1) results in

$$0.93 \leq \frac{\partial(\Delta R_{\text{cf}}(p))}{\partial p} \leq 3.042 \tag{4.8}$$

Consequently, the control law (4.6) guarantees $\Delta R_{\text{cf}} \to \Delta R_{\text{cf}}^{ref}$ and can be rewritten as

$$p = -K \int (\Delta R_{\text{cf}}(p) - \Delta R_{\text{cf}}^{ref}) dt \tag{4.9}$$

This is an integral control, i.e., I-only controller with the parameter $K > 0$ which can be tuned to achieve a desired performance in keeping $\Delta R_{\text{cf}} = \Delta R_{\text{cf}}^{ref}$. Although this controller is not a part of our proposed control architecture, the presented analysis together with (4.1)-(4.3) inform our control design.

4.4 Control Architecture

Our control architecture has two main parts, see Fig. 4.1. The first one is the reference tracking controller for the curvature or, more precisely, for the curvature sensor resistance and the second one is the force controller. In our control architecture for the firm touch, we start with the reference tracking controller. This controller moves the SSA, i.e., increases its curvature and is active until the SSA is in contact with an object in its way. Then the control of the SSA is switched to the force controller to achieve a firm touch with the object.

The idea behind the use of the reference signal to set the SSA in motion is that we know that if the SSA moves freely, then the error of tracking the reference is small. For example, if the reference signal is periodic, then during the one period of the signal, the free moving SSA will move as illustrated in Fig. 4.7a-e. This motion resembles a finger that samples points in space to touch and is also an integral part of the opening and closing of multi-finger grippers. If the SSA eventually touches an object, (see Fig. 4.7f-h) the SSA movement is not free and the error of tracking the reference for the resistance of the curvature sensor will increase. We can measure that error and once its value exceeds a certain threshold, we can robustly conclude that the SSA is in contact with the object and we switch to the force controller.

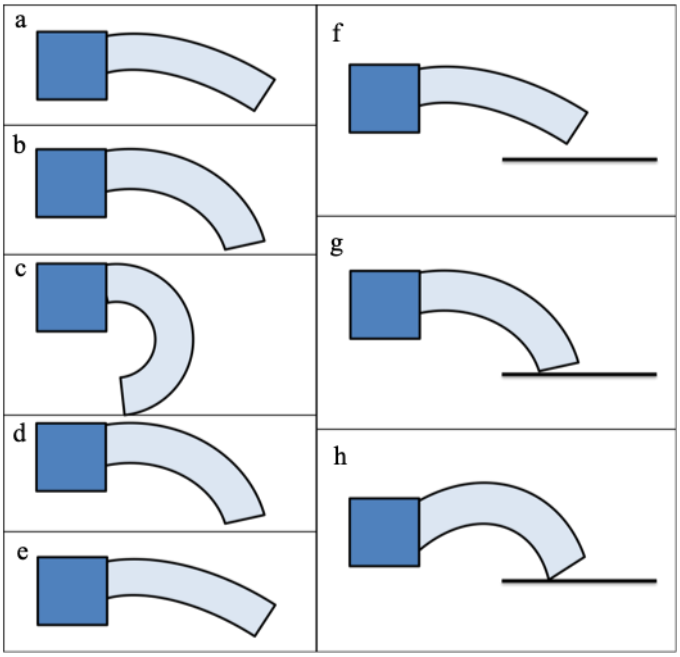

Figure 4.7: The left side, a-e, is an example of how the SSA configuration changes in the free motion as the inflation pressure increases and then decreases. The right side, f-h, is an example of how the SSA configurations change if the motion is blocked and the inflation pressure increases.

The block that compares the error with the threshold T is depicted in Fig. 4.1 and it has the set and reset inputs. Before we start the movement, i.e., activate the curvature tracking controller $D_c(s)$, we reset the output of the block. Once the threshold is reached and

$$\Delta R_{\text{cf}}^{ref} - \Delta R_{\text{cf}} > T \tag{4.10}$$

the output of the block is set to 1 and the inflation pressure control is switched to the force controller $D_t(s)$.

4.4.1 Reference Tracking Curvature Control

The Lyapunov function analysis in Section 4.3 shows that an I-only controller is sufficient for the curvature control and indicates that a PI controller may be needed for an improved response of the control loop. However, from the linear system control theory, we know that such a controller cannot guarantee that the error of tracking references goes to zero for reference signals that are different from the step function.

We consider here the controller for tracking references that are composed of the so-called ramp signals. Therefore, based on the linear theory to achieve the convergence of the tracking error to zero, we have to consider the following 2nd order linear controller

$$D_c(s) = K_c \frac{s + z_c}{s^2} \tag{4.11}$$

which is given in the form of a transfer function, where s is the Laplace domain variable. The controller parameters are gain K_c and the so-called zero of the transfer function z_c. The characteristic polynomial of the linearized closed loop control for the curvature control is

$$L_c(s) = 1 + D_c(s) \frac{\partial(\Delta R_{\text{cf}}(p))}{\partial p} = 1 + K_c \frac{s + z_c}{s^2} \frac{\partial(\Delta R_{\text{cf}}(p))}{\partial p} \tag{4.12}$$

in which $\frac{\partial(\Delta R_{\text{cf}}(p))}{\partial p} \in [0.93, 3.042]$ varies as a function of the inflation pressure p, which is the control variable, see Section 4.3.

Given $L_c(s)$, we used the root locus analysis to analyze and tune the parameters of the controller so the gain, K_c, was within the possible range of pressure, and the zero, z_c, was chosen to have reasonable time response for reference signal tracking. We ultimately decided to use $z_c = 2$, $K_c = 50$ and illustrated the parameter selection with the simulation result in Fig. 4.8.

In the simulation of the controller, we chose a periodic triangular signal with the peak value corresponding to the maximal measured value in the data, see Fig. 4.4. The period of the reference is 5 seconds, which dictates that the SSA completely flexes and extends. This is a relatively slow motion for the SSA, but it is decided upon intentionally, so that we can base our control design without taking into account higher frequency modes of the SSA's dynamics. Fig. 4.8 shows that the controller designed for tracking the curvature of the finger follows the reference input very closely, it only differs at sharp changes, which is to be expected, and the tracking error settles to zero between direction changes.

4.4.2 Force Control

Based on the Lyapunov function analysis in Section 4.3, we know that a PI controller is sufficient to follow the reference input of a step if the system we control is sufficiently well modeled by a nonlinear characteristic such as (4.2). For the force control, we use the step reference input because of the intention to establish the controlled force between the SSA and an object quickly after their contact is detected.

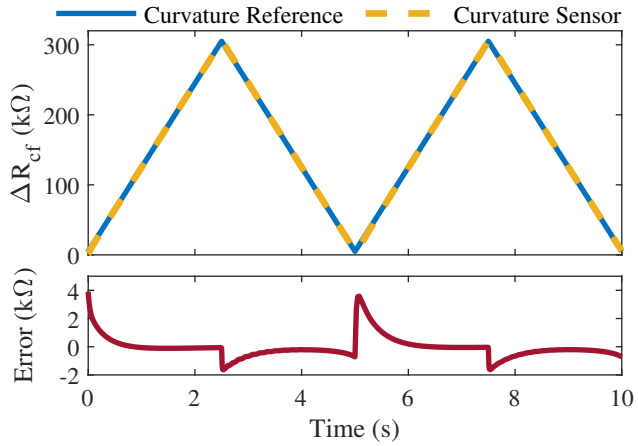

Figure 4.8: The curvature tracking controller: the reference input is the triangular signal that results in the flexing and extending of the SSA depicted in blue, the result of the curvature sensor tracking the reference is in yellow and overlaps with the reference input, and the the tracking error is in red.

The Laplace transfer function of the PI controller we consider here is

$$D_t = K_p + \frac{K_i}{s} \tag{4.13}$$

$$= K_t \frac{s+z_t}{s} \tag{4.14}$$

with the control parameters K_t and z_t. This controller yields the characteristic polynomial of the closed loop for the force control as

$$L_t(s) = 1 + D_t(s)\frac{\partial(\Delta R_{\text{to}}(p))}{\partial p} = 1 + K_t \frac{s+z_t}{s}\frac{\partial(\Delta R_{\text{to}}(p))}{\partial p} \tag{4.15}$$

in which based on (4.2), we have $0.01738 \leq \frac{\partial(\Delta R_{\text{to}}(p))}{\partial p} \leq 8.6410$.

Using the root locus analysis, we played with K_t and z_t to obtain $K_t = 50$ and $z_t = 5$ such that the gain was within the possible range of pressure and the zero chosen to have reasonable time response characteristics, and illustrated results with the simulation in Fig. 4.9. In the simulation, the amplitude of the step was chosen based on the corresponding higher inflation pressure, which is important for a firm contact with the object.

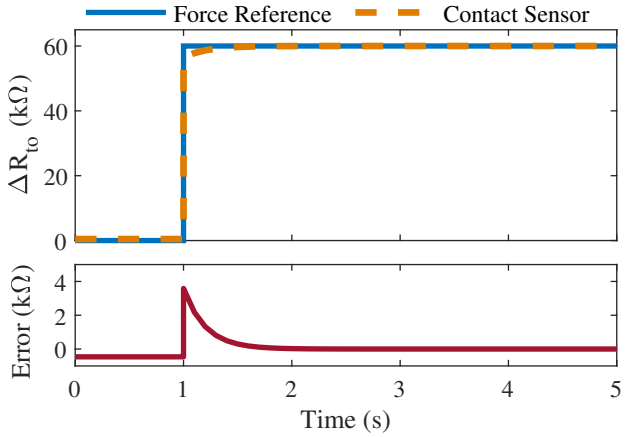

Figure 4.9: The force controller: the reference input for keeping the constant force is depicted in blue, the output, which is the contact sensor resistance, is in orange, and the control error is in red.

4.5 Results

We present in this section the simulation resulting from the proposed control strategy and using the control parameters from the previous section. The SSA's curvature and contact sensor signals are modeled with the pressure dependent relations (4.1) and (4.2). The threshold $T = 5\text{k}\Omega$ (see expression (4.10)) is determined as a value that is higher than any value of the curvature reference tracking error plotted in Fig. 4.8, which results from tracking the triangular reference while the SSA is free from any contact.

To simulate a firm touch with an object, a disturbance on the SSA curvature sensor is introduced. The disturbance is additive in the form of a decrease in resistance, which accounts for a change that the curvature sensor resistance as a function of pressure would hypothetically experience during the blocked displacement. This disturbance is introduced as the SSA flexes when reaching the object and is maintained for the remainder of the simulation to simulate steady,

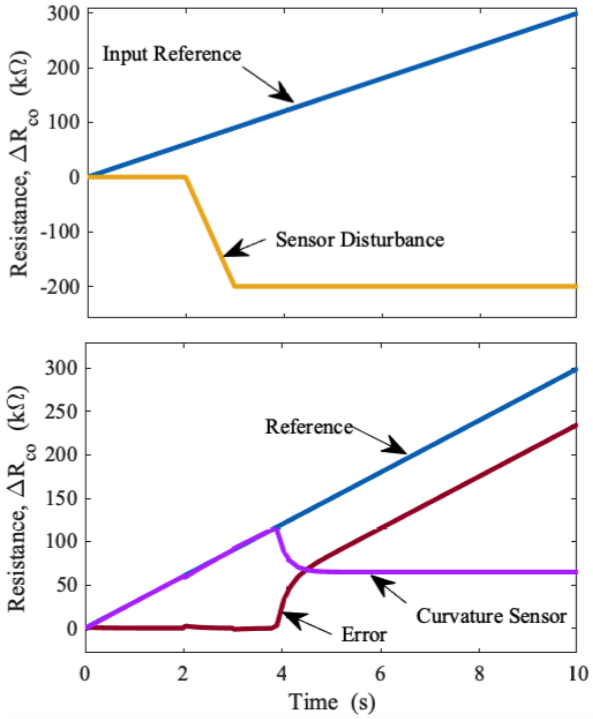

Figure 4.10: The top plot shows the reference input (blue) which results in the SSA flexing. The force disturbance (yellow) simulates the SSA coming into contact with an object. The bottom plot shows the result of the curvature tracking controller under the disturbance. The tracking error is in red and the contact sensor resistance is in orange.

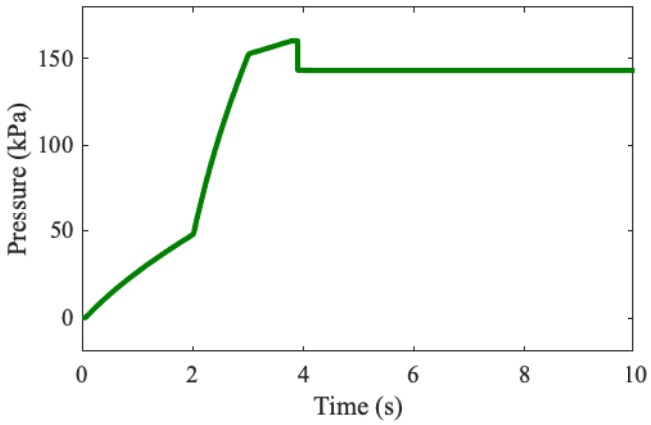

Figure 4.11: The output of the feedback controller that actuates the SSA.

firm contact with the object.

The top plot in Fig. 4.10 shows the ramp reference required for the curvature controller to set the free moving SSA in motion and the disturbance used to create the decrease in the curvature sensor's resistance once in the blocked displacement. The bottom plot in Fig. 4.10 shows the reference input, the impact of the disturbance on the curvature sensor, and the reference tracking error. At the time $t = 2$ seconds, the disturbance begins to be introduced, simulating contact with the object. This creates an increase in pressure since the finger has changed from the free to the blocked displacement, which can be seen in Fig. 4.11. As the pressure increases, the finger continues to bend. However, since the finger is now in the blocked displacement, the curvature sensor is not able to track the reference input. The divergence from the reference input causes the error to increase, which gives the overall controller the signal that the finger has a firm touch with the object, and the control switches from the curvature to the force control. The top plot in Fig. 4.12 shows that the error passes the threshold around the time $t = 3.95$ seconds, which is indicated in the bottom plot of Fig. 4.12. This plot shows the curvature sensor resistance before and the contact sensor resistance after the switch time at $t = 3.95$ seconds. After that point, the force controller maintains the resistance of the contact sensor to sustain the firm contact with the object.

4.6 Conclusion

In this paper, we presented the control for the SSA to achieve a firm contact with an object. The control is based on the curvature reference tracking and force controllers. The

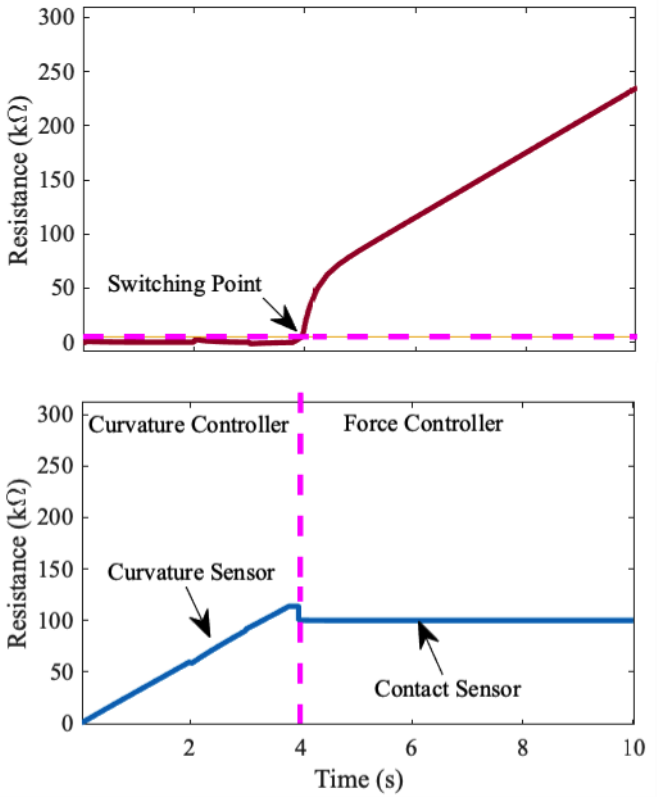

Figure 4.12: Switching point and outputs of the two controllers: the top plot illustrates the event of the curvature tracking error (blue) passing the threshold (red). This time point of the event is illustrated with the dashed line on the bottom plot. The blue line on the bottom plot shows the resistances of the active controllers. The curvature tracking controller is active at the time before the dashed line and the force control is active after the dashed line.

tracking controller sets the SSA in motion which becomes impeded if the SSA is in contact with an object on its way. This increases the tracking error and once its level passes a threshold, the control switches to the force controller. Each controller's structure and parameters were developed taking into account both the unique features of the problem and experimental data. The results were tested in the simulations to prove the concept. The future work is aimed at comparing the simulation data with signals recorded in experiments with the controlled SSA. This comparison will give a better insight into the dynamics of the SSA from data recorded in closed-loop experiments, which will allows us further progress towards developing control algorithms for the control of the SSA-based multi-finger robotic grippers and anthropomorphic robotic hands.

Chapter 5

Controlled Multilayered Soft Actuator Containing Embedded Intrinsically Soft Sensors

This chapter is a preprint to the paper

- M. Boivin, C. Conrad, M. Wehner, and D. Milutinović, "Controlled Multilayered Soft Actuator Containing Embedded Intrinsically Soft Sensors," (in prep).

5.1 Introduction

The development of a controllable finger-like soft robotic component is a prerequisite for the development of robust, controllable multi-finger soft robotic grippers or hands. Inspired by humans who use proprioceptive and haptic sensing, we present here the design of a finger-like soft robotic component with embedded sensors (Fig. 1A and B). The sensors are the curvature

sensor along the dorsal ridge of the finger and the force sensor at the finger's tip. *We use this finger as a platform to introduce a novel control enabled touch detection method, which does not require the force sensor*.

The proposed detection method overcomes the necessity for the co-location between the point of touch and the corresponding sensor. Many existing soft fingers require a precise positioning of an object directly in contact with the most sensitive point of the force sensor, reducing the effectiveness in real world scenarios. Moreover, while we have presented our method on a specific soft finger, our method can be used on a wide variety of soft robotic fingers with curvature sensors [55, 89, 128, 129]. Once the finger is in touch with an object, the finger stops moving due to the contact with an object. In our approach, the pneumatic inflation of the finger continues, increasing the contact force between the finger and the object. It also results in deviating the pose of the finger from its free-displacement state until the touch is detected. This is fundamentally different from traditional rigid robots (where an increased torque would cause a minimal deflection of the rigid actuator and would likely damage the object or motor). After the touch is detected, the finger is in the state of a *compliant touch with an object* and achieving this state is a prerequisite for controlled grasping with a soft robot gripper. Note that due to the compliant nature of soft fingers, it is less crucial for the generalized grasping strategy [55] to limit contact force, as the gripper will simply deflect rather than crushing the object.

Soft robotic components show promising results in areas like grasping [110] and biomedical applications [25] through various forms of locomotion and sensing [67]. However, soft compliant robot components have less constrained movements which can be unpredictable and difficult to control [105]. This unpredictability can result in unexpected configurations

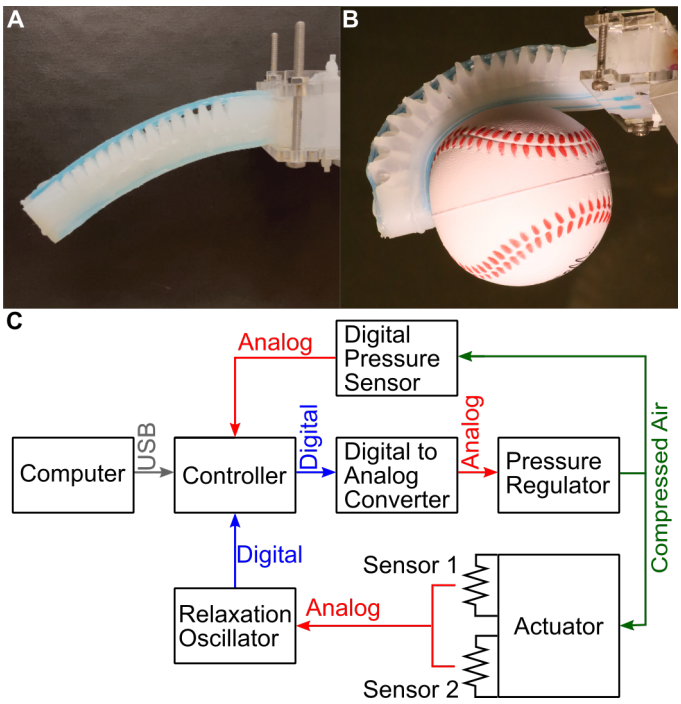

Figure 5.1: **The soft somatosensitive actuator (finger) design. (A)** The finger is deflated. **(B)** The finger is inflated and holding onto a foam ball to show how it curls around objects. **(C)** The complete system setup needed to collect data and control one of the fingers. The central component of the system is the Arduino Nano which is used for both controlling the finger and collecting data from the sensors. These data are then sent via serial port to the computer.

at a given time, especially when the component is in contact with objects in the operating environment [55]. The mechanical finger design, embedded proprioceptive sensors [24], or exteroceptive sensors on soft robotic components can greatly enhance the ways in which finger-like actuators interact with objects [110] and how they can be controlled. In this respect, finger-like actuators are part of a general trend of the larger area of soft robotics [99, 102, 120, 122].

Finger-like actuators are often equipped with a sensor closely following the current shape of the actuator and measuring its bending, i.e., its curvature. These sensors can be either off-the-shelf curvature sensors embedded in the actuator's body [45, 55] or embedded sensors created within the finger. The latter are based on a variable resistance [117, 118], magnetic [95] and optical sensors [129]. Many soft finger actuators include a force sensor [119] embedded in or attached to the finger's tip to detect pressure or contact with another object [56, 117, 118].

Inspired by the finger-like actuator [118], the soft finger presented here has embedded resistance based curvature and force sensors, see Fig. 2A. Both sensors are soft and capable of stretching with the soft finger's body. We connect the soft finger to a pressure regulator and electronics hardware to create the validation system, schematically represented in Fig. 1C, which we use to introduce and test our control enabled touch detection method. Although the force sensor is present in our system, it is not required for our method, freeing it for other uses such as haptics or temperature sensing.

Kinematics and dynamics of soft robotic components are typically highly nonlinear which is also reflected in the complexity of control [43]. One approach to coping with it is to use data-driven techniques, like neural networks [20, 114], reinforcement learning [113], or a combination of data-driven and model-based approaches [85, 116]. Another approach is to

simplify models taking into account the range and dynamics of motion [124], which has been successfully used for soft actuators with multiple soft segments [32] and with motion capturing touch detection [61]. In the case of finger-like actuators with a single soft segment, linear PID controllers can be used to achieve position control [45, 89, 128]. The existing works on control report the use of proprioceptive sensing for position control [45, 89, 128] or control for a compliant motion of the finger [61] using a motion capturing system.

In this paper, we rethink the role that feedback control can play in the control of soft finger-like actuators. Instead of using feedback control to control precisely the position or motion of a soft finger, we propose to use feedback control for touch detection. In our method, we design a linear feedback control loop in which a curvature sensor resistance tracks a reference signal. If the finger flexing is unimpeded, then the reference tracking error is negligible. However, if the flexing of the finger is impeded by a contact with an object, then the tracking error increases. This increasing error is used to detect that the finger is in touch with the object. Once the touch is detected, the reference signal is changed from a free-displacement reference to a blocked-displacement reference, ensuring that the finger remains in the compliant touch with the object.

The notable contribution of our work is the use of a feedback control loop not to precisely position the finger, but to obtain the closed-loop controlled curvature sensor resistance and use the resistance reference tracking error to detect the touch with the object and maintain contact with it. A preliminary numerical simulation-based version of a portion of this work appeared in the conference paper [13], which introduced the idea of using a reference tracking controller for soft robotic fingers. The work presented here greatly extends that simulation-based

work and validates the concept through experiments on a physical device.

As part of the validation, we provide (1) details about our manufacturing method with embedded liquid sensors and other components of our test system, including electronics and air pressure control. Our system can be constructed without specialized or expensive materials or equipment. Then we provide (2) the characterization of our embedded sensors as useful information for a comparison between our and similar physical platforms. Finally, we use (3) the reference tracking feedback control loop of the curvature sensor resistance and the resistance tracking error in our touch detection method. The design of the feedback control highlights that our results are due to the applied method, rather than a sophisticated analytical or data-driven model of the finger.

The paper is organized as follows. The following *Materials and Methods* section presents the resulting soft sensor-actuator system design, the fabrication of the soft finger, and design verification with sensor characteristics followed by the closed-loop reference tracking control of the curvature sensor resistance. The *Results* section presents the validation of the closed-loop reference tracking for touch detection. It also includes a summary of the results. The paper finishes with the conclusion section.

5.2 Materials and Methods

The physical device used in this work is a pneumatically actuated soft finger containing embedded ionically conductive fluidic sensors (Fig. 2A). The finger is made out of multiple materials (Fig. 2B) to maximize grip strength without sacrificing sensor responsiveness. In our

experiments, the finger was mounted to a frame where it was instrumented with pneumatic and electrical connections. Next, we describe features of all components from our validation system (Fig. 1C).

5.2.1 Soft Finger Design Features

Our work presented here requires that a soft finger contains at least one integrated curvature sensor and the design of the finger we use (Fig. 2A) is inspired by the work of Truby et. al. [118]

In addition to a network of pneumatic bladders for actuation, the finger contains a *curvature sensor* along the length of the dorsal ridge (Fig. 2A). We use this sensor in our touch detection approach. The other sensor, the *force sensor*, is not used in our approach and can be used for other haptic tasks (sensing contour or temperature).

The finger body layers are constructed of two kinds of silicone elastomers, *Ecoflex-0030* and *Dragon Skin 10* as depicted in Fig. 2B. The curvature sensor layers were constructed entirely of *Ecoflex-0030* to maximize deflection and elongation with inflation pressure, while the force sensor layers were constructed of *Dragon Skin 10*. The finger bladder assembly is constructed of *Dragon Skin 10* and adhered to a layer of the same material. Upon inflation, this non-homogeneous assembly yields asymmetric actuation, and results in the finger bending.

Sensor channels were filled with the ionic fluid 1-ethyl-3-methylimidazolium ethyl sulfate (1E3MES) [118]. The channels were extended to the proximal end of the assembly, and terminated within the frame-mount system in order to allow secure interface with electrical connectors leading to the controller.

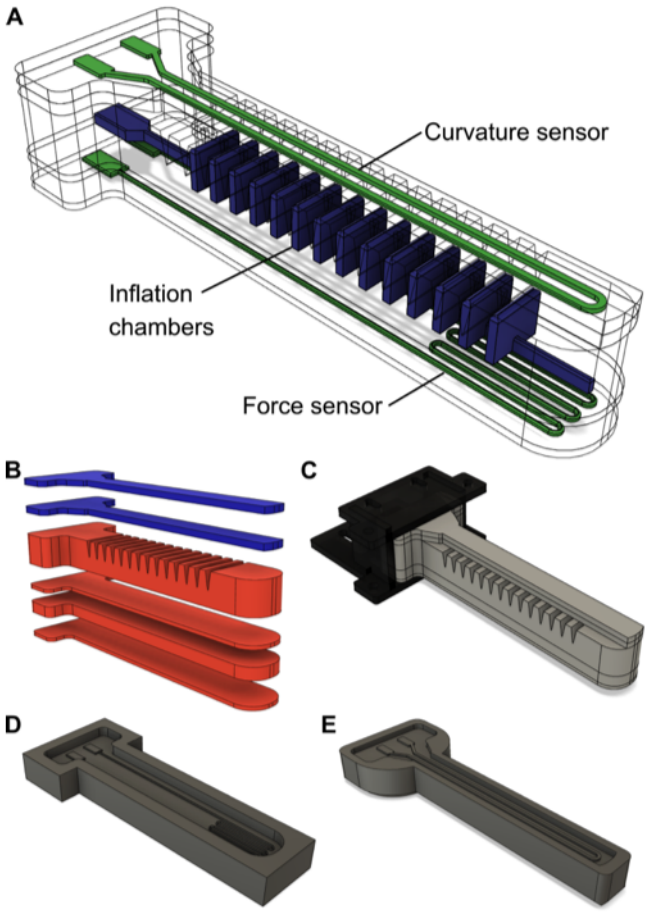

Figure 5.2: **Soft robotic finger with embedded soft sensors model. (A)** The schematic of the complete finger that was developed. The curvature (top) and force (bottom) sensors are shown in green, and inflation chambers in blue. **(B)** Elastomer layers of the finger that was developed. It features multiple types of silicone rubber which allow it to retain sensitivity while exerting more force. The layers in blue are made out of Ecoflex 00-30 while the layers in orange are made out of Dragon Skin 10. **(C)** A mount was created to reduce stress and prevent the wires and the air supplying needle from being pulled out. The mount is designed to be filled with uncured silicone behind the finger to fully lock in the components. **(D)** The bottom force sensor mold is filled with Dragon Skin 10. **(E)** The top curvature sensor mold is filled with Ecoflex 00-30, so the material remains easily flexible.

The general fabrication method for both sensors is the same. The sensors have the same working principle of varying electrical resistance with changes in geometry. A channel of an initial geometry (length and cross-section) exhibits an initial resistance. If that channel is stretched, the resulting geometric change would increase resistance (greater length, smaller cross section area). If an external force compresses a portion of a channel, the local cross section will decrease and electrical resistance will rise, allowing us to infer pressure from the resistance change. Various sensing modes can be obtained utilizing this correlation between geometry and resistance, and designing sensor channels accordingly.

We chose the associated names for each sensor based on the intended use. The curvature sensor was placed along the dorsal edge of the soft finger, where it stretched with bending (because it is on the far side of the neutral axis of bending). The force sensor was placed on the underside of the finger to measure the force upon a contact with an object. The force sensor could also be used for a general cutaneous sensing, i.e. tactile or temperature sensing. The complete fabrication process used to create the soft finger is provided in the Supplementary Materials.

5.2.2 Validation System

The schematics of the validation system is depicted in Fig. 1C. An Arduino Nano outputs a control signal (through a digital-to-analog converter) to a digital pressure regulator analog input, which sets the air pressure supplied to the soft finger. The pressurized air results in the finger bending, which deforms the curvature sensor, changing both length and cross section area, thus changing resistance in the contained ionic gel sensor. This resistance is a part of a

relaxation oscillator depicted in Fig. 3 and represented with R_s. The relation between R_s and the oscillator frequency f_s, i.e., its period T_s is

$$f_s = \frac{1}{T_s} = \frac{1}{2R_sC\ln\frac{1+a}{1-a}}, \tag{5.1}$$

where $a = R_1/(R_1+R_2)$, $R_1 = 680\ \Omega, R_2 = 9.9\ k\Omega, C = 10\ nF$. The oscillator generates a symmetric voltage signal across R_s without a DC component, which prevents the electrolysis of the ionic fluid 1E3MES and the sensor damage [23]. Each sensor of the finger has its own relaxation oscillator of the same type. The frequency of the signals are measured by the Arduino using interrupt and timer services.

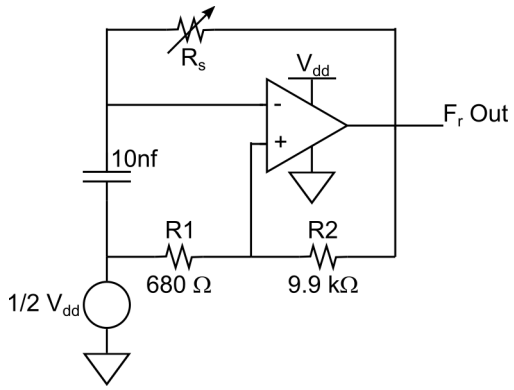

Figure 5.3: **The relaxation oscillator circuit.** The circuit used to measure the sensor fluid resistance. The relaxation oscillator outputs a variable frequency square wave based on the sensor resistance R_s.

5.2.3 Curvature Sensor Resistance Characteristics

To characterize the finger assembly, we mounted it horizontally, so that it could bend freely without disturbance, as shown in Fig. 4A. The initial tilt of the finger due to the effects of gravity was a result of the finger's nonrigid nature (shown in Fig. 1A). The digital pressure

regulator added some oscillatory noise to the system as its valves opened and closed, and had a delay in response to the commanded pressure, which in turn affected the response of the curvature sensor resistance, see Fig. S1-S2 in the Supplementary Material.

In performing the test, pressure for the regulator started at 0 kPa and was increased in increments of 0.7 kPa (0.1 PSI) to 50 kPa (7.25 PSI). Each pressure state was held for 15 seconds to capture the variation in sensor measurement. To prevent rupture, the pressure was limited to 50 kPa.

The change in resistance, $\Delta R_{\mathrm{cf}}(p)$, is given by $\Delta R_{\mathrm{cf}}(p) = R_{\mathrm{cf}}(p) - R_{\mathrm{cf}}(0)$, where $R_{\mathrm{cf}}(p)$ is the curvature sensor resistance for the pressure p and $R_{\mathrm{cf}}(0)$ for the pressure $p = 0$. We found a polynomial approximation to the characterized data of the free moving finger (Fig. 4B) to obtain an approximate empirical relation as

$$\Delta R_{\mathrm{cf}}(p) = 4.964 \cdot 10^{-5} p^4 - 0.002687 p^3 + 0.05297 p^2 + 0.7648 p + 0.1988, \quad (5.2)$$

where p is in kPa and $\Delta R_{\mathrm{cf}}(p)$ in kΩ. The function $\Delta R_{\mathrm{cf}}(p)$ is a monotonously increasing function and its first derivative with respect to p is

$$\frac{\partial(\Delta R_{\mathrm{cf}}(p))}{\partial p} \in [0.1988, 10.7293]. \quad (5.3)$$

If the soft finger is in contact with an object as in experiments depicted in Fig. 4C, 4E and 4G, we see that the relation between the pressure and the change of resistance is different as can be concluded from Fig. 4D, 4F and 4H, respectively. In the last case, we obtained an

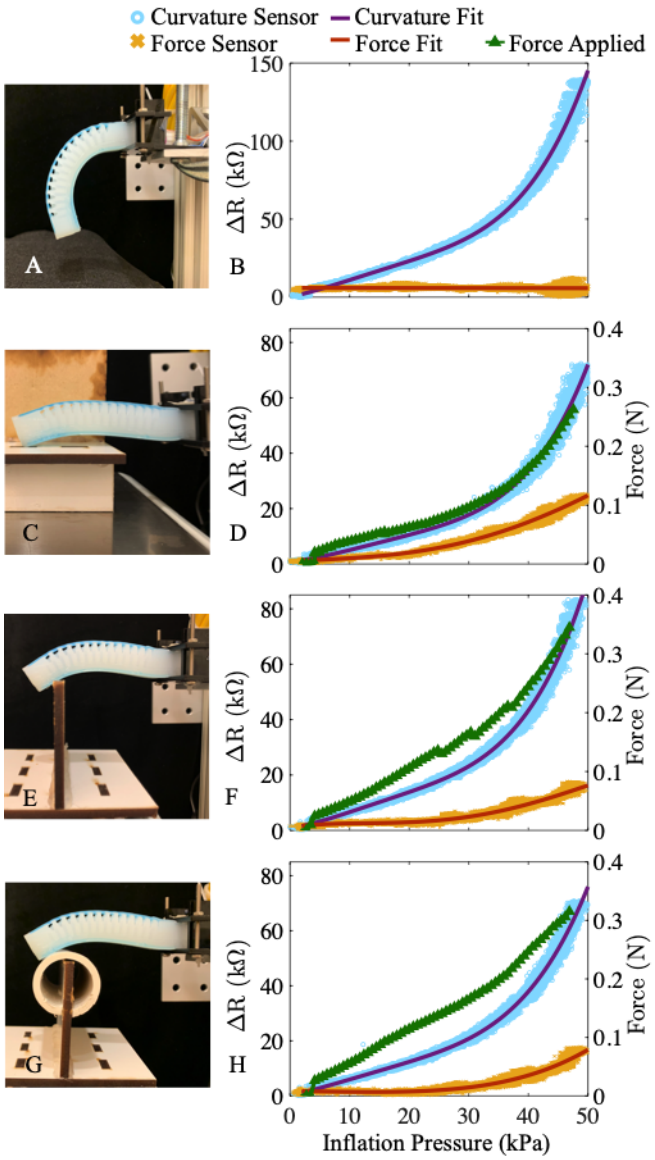

Figure 5.4: **Free and Blocked displacement tests for characterizing the sensors.** **(A)** The finger deflated (left) and inflated (right) in free displacement. **(B)** The corresponding data for free displacement. **(C)** The finger inflated on a flat surface. **(D)** The corresponding data for blocked displacement on a flat surface. **(E)** The finger inflated on a post for a fine contact point. **(F)** The corresponding data for blocked displacement on a post. **(G)** The finger inflated on a round object. **(H)** The corresponding data for blocked displacement on a tube. For all data shown, the curvature sensor change of resistance is in blue with the fit to data shown in purple and the force sensor in gold with the fit to data shown in dark orange. The applied force is shown in green.

empirical relation

$$\Delta R_{\text{co}}(p) = 2.27 \cdot 10^{-5} p^4 - 0.001145 p^3 + 0.02184 p^2 - 0.4481 p + 0.2213, \quad (5.4)$$

where $\Delta R_{\text{co}}(p) = R_{\text{co}}(p) - R_{\text{cf}}(0)$ and $R_{\text{co}}(p)$ is the resistance of the curvature at the pressure p while the finger is in contact with the cylinder. The relations for the contacts in Fig. 4C and 4E are different, yet similar enough to be written as the 4th order polynomials with the positive derivative of ΔR_{co} with respect to p. To test the repeatability of our results, an additional soft finger with embedded sensors was characterized using the same process and the plots are shown in Fig. S3. All of the relations were fitted with the 4th order polynomials and the positive derivative of ΔR with respect to p. The significance of this was illustrated by the preliminary analysis presented in our previous work [13].

5.2.4 Force Sensor

Many soft robot fingers include a dedicated force sensor at their distal end which directly measures contact force or pressure between the finger and an object being manipulated. While our method does not require the sensor in order to function, we have included a force sensor at the distal end of the finger. We use this sensor to verify that the finger is in contact with an object in the validation of our touch detection method (see Fig. 7D and 7F). In future systems, this force sensor could either be eliminated or used for other purposes such as temperature sensing or haptics. To characterize the force sensor, we performed experiments with the finger in contact with an object. For each experiment, the soft deflated finger was mounted, so that it

rested on an object (see Fig. 4) that was placed on top of a scale for measuring force, and then the finger was pressurized up to 50 kPa (7.25 PSI) using a digital regulator. The top curvature sensor resistance and the bottom force sensor were recorded. The force value on the scale along with the pressure was captured through video data.

The first experiment consisted of placing the finger in contact with a flat surface and the inflation pressure was increased holding at each pressure for 15 seconds; this setup and the results are shown in Fig. 4C and 4D. The finger was placed so that at least half of it rested on the flat surface. The finger tended to lift when on a flat surface due to its design and it curved around an object. However, there was still an overall change in resistance, $\Delta R \approx 20\ k\Omega$, and an applied force of approximately 0.25 N across inflation pressure.

The next test was performed with the finger in contact with a post (see Fig. 4E and 4F). The post was chosen because it would allow the finger to apply force over a much smaller area. The post was placed so that when the finger was at rest, the force sensor channel at the tip of the finger was centered on the post. Fig. 4E shows that the finger's tendency was still to curve around the post, but the post remained in contact with the force sensor. Just below 30 kPa, an increase in sensor response was noticed and the applied force increased to approximately 0.35 N across inflation pressure.

The final test had the pressurized finger placed against a cylinder, as shown in Fig. 4G and 4H. This test was performed in the same way as the previous tests and had the cylinder centered under the force sensor. Overall, the results were similar to the force test on the post.

5.2.5 Control Architecture

The controller in Fig. 5 is a curvature sensor reference tracking controller. The work of this controller results in the curvature sensor change of resistance ΔR_c values to follow the reference ΔR_c^{Ref} signal. This is equivalent to the resistance R_c values to follow R_c^{Ref} reference signal since $\Delta R_c = R_c - R_c(0)$. For a correctly designed reference tracking controller and freely moving finger, we expect a small tracking error $R_c^{Ref} - R_c$. Therefore, the tracking error signal can be used to detect contact of the finger with an object in the operating environment [13]. A periodic reference signal sets the freely moving finger in a motion that resembles the motion of a finger that samples points in space to touch. Similar types of motion are also an integral part of the opening and closing of multi-finger grippers.

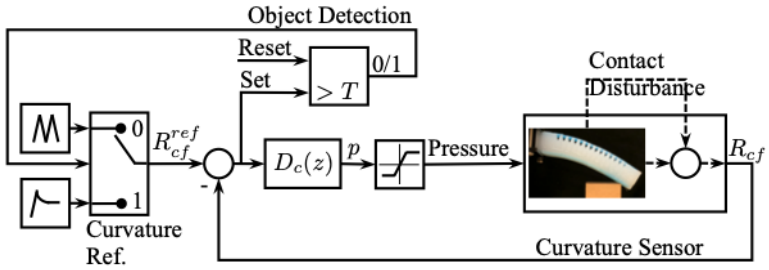

Figure 5.5: **The block diagram of the feedback control loop**. The controller D_c outputs pressure p, which results in the curvature sensor resistance R_c tracking of the reference values R_c^{ref}. Once the tracking error exceeds a threshold, the periodic signal R_c^{ref} is changed to a reference signal which maintains the contact with an object.

5.3 Results

The characterizations of the curvature and force sensor (see Materials and Methods) show that both sensors are responsive and have potential in applications of the soft finger.

However, the signal level of the force sensor strongly depends on the collocation between the force sensor position and the point of contact with an object. Because of that, our feedback control enabled touch detection method (see Materials and Methods, Control Architecture and Fig. 5) relies only on the curvature sensor.

It is important to note that we decided to forgo the process of mapping the position and angle of the tip of the finger since our control architecture utilizes the error in the curvature sensor resistance when tracking a reference. The control architecture is designed so that it is not focused on a specific position or angle, but on whether the finger can detect when it has made contact with an object.

A critical part of our control architecture is the reference tracking feedback controller depicted as D_c in Fig. 5, which we present and validate in the next subsection. This is followed by the validation results of the feedback control enabled touch detection proposed in this work.

5.3.1 Design and Validation of Reference Tracking Curvature Control D_c

Our previous work [13] shows that for a soft finger with a curvature sensor characterized by the polynomial approximation (2), *the reference keeping* control can be achieved by

$$p = -K \int (\Delta R_{\mathrm{c}}(p) - \Delta R_{\mathrm{c}}^{ref}) dt. \tag{5.5}$$

This is an integral control, i.e., I-only controller with the parameter $K > 0$ which can be tuned to achieve a desired performance in keeping $\Delta R_{\mathrm{c}} = \Delta R_{\mathrm{c}}^{ref}$. A further performance improvement could be achieved by a PI controller. However, from linear system control theory, we know that in our system such a controller cannot guarantee that the error of tracking references goes to zero

for reference signals that are different from the step function.

We consider here the controller for tracking references that are composed of ramp signals. To account for that and achieve the convergence of the tracking error to zero, we consider the 2nd order linear controller

$$D_c(s) = K_c \frac{s + z_c}{s^2}, \tag{5.6}$$

which is given in the form of a transfer function, where s is the Laplace domain variable. The controller parameters are gain K_c and the so-called zero of the transfer function, z_c. In our controller implementation we used an Arduino Nano computer platform. Therefore, for the implementation we convert the transfer function (7) to the discrete time domain as

$$D_c(z) = K_c \frac{z + z_c}{(z-1)^2} \tag{5.7}$$

This transfer function, after the forward approximation method sample-hold step [9], replaces $D_c(s)$ in Fig. 5. In that case, the steady state of the error $e_{ss} = \lim_{t\to\infty} e(t)$ for tracking the ramp signal reference is

$$e_{ss} = \lim_{z\to 1}(z-1)\frac{1}{1 + D_c(z)\frac{\partial(\Delta R_{\text{cf}}(p))}{\partial p}} \Delta R_{\text{cf}}^{ref}(z). \tag{5.8}$$

For the zero error tracking of a unit ramp signal reference which has the z-transform $hz/(z-1)$, where $h = 0.2s$ is the sample time, we need $\lim_{z\to 1}(z-1)D_c(z)\frac{\partial(\Delta R_{\text{cf}}(p))}{\partial p} = \infty$. We can confirm that due to (3), i.e., the bounded derivative $\frac{\partial \Delta R_{\text{cf}}(p)}{\partial p}$, the controller $D_c(z)$ satisfies that condition, but we have to tune the gain K_c and place the zero z_c of the controller. To tune these parameters, we use the discrete root locus analysis with criteria of the damping ratio of $\zeta = 0.707$ on the root locus diagram while also trying to optimize rise and settling times in the step response. From these, we selected $K_c = 0.07$ and $z_c = 0.955$.

For testing the controller, we chose a periodic triangular signal where the range for the reference signal was chosen based on the data we collected to characterize the curvature sensor (see Fig. 4). The period of the reference was approximately 5s, and during that time the finger completely flexed and extended. This period allowed us to use the simple controller and was a safeguard against higher frequency modes of the system's dynamics. The finger was mounted on a test stand and placed in free displacement for validating the reference tracking curvature sensor resistance controller.

The performance of the controller is illustrated by data in Fig. 6. The controller worked for multiple orientations of the finger. In Fig. 6A the finger was curling down toward the ground, in Fig. 6B the finger was rotated on its side, and in Fig. 6C it was curling up toward the ceiling against gravitation. For all of these configurations, the controller designed for tracking the curvature of the finger followed the reference input closely. As expected, the tracking error was larger around time points at which the reference slope changed its direction. The tracking error settled around zero between direction changes. For the case of the finger curling up, the resulting control data showed that the finger struggled to curl to the maximum reference. This was due to the sensor being in an orientation in which gravity acted against the direction of curvature of the inflated finger in the sagittal plane. This meant that the curvature sensor resistance range shifted, so more pressure was required to reach a curvature resistance reference that was designed for when the finger curled down with the help of gravity. However, to prevent the inflation chambers from rupturing, there is a limit to the amount of pressure that can be applied. Due to the limits on the amount of pressure that the finger could withstand, anti-windup was included to stop the integration and accumulation of error when reaching the saturation limits [9]. The control data

presented in the final plot, see Fig. 6D, tested the limit of the maximum pressure and showed that the anti-windup designed to stop integration worked to keep the finger from becoming unstable.

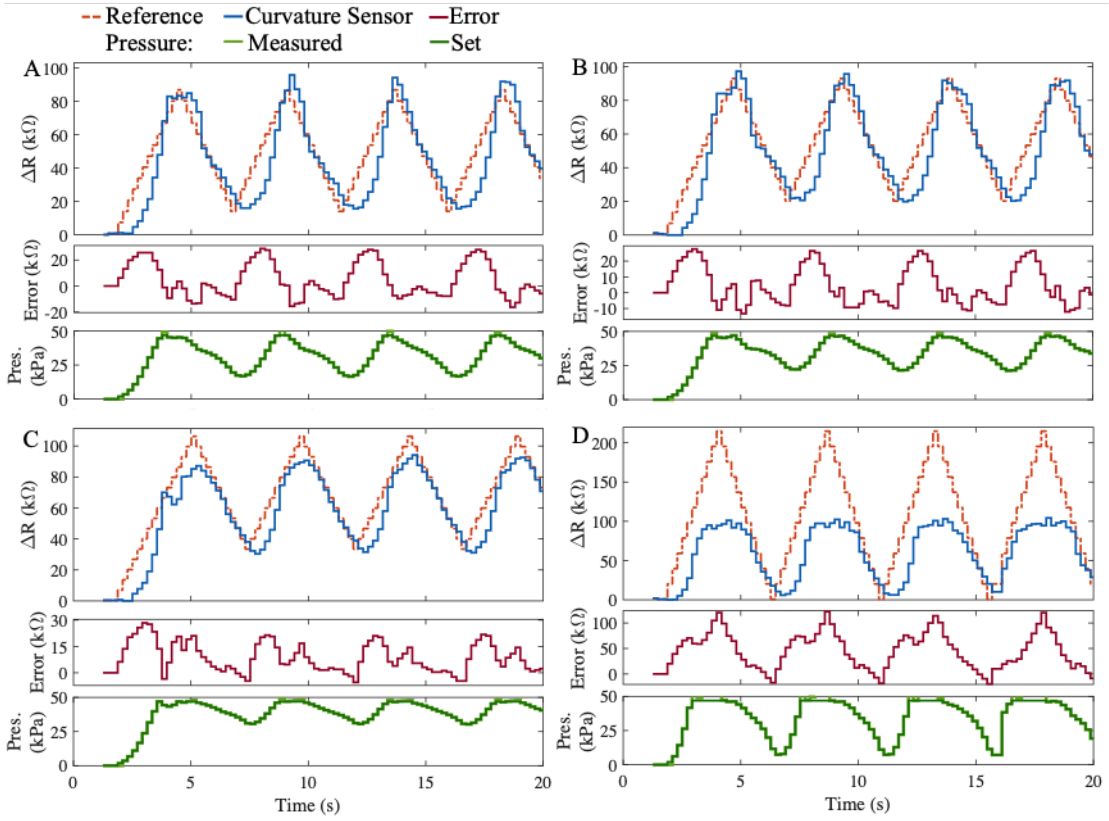

Figure 5.6: **The curvature sensor resistance tracking controller for different orientations of the finger.** In all four plots, the top section shows the reference signal as $\Delta R_{\mathrm{c}}^{ref} = R_{\mathrm{c}}^{ref} - R_{\mathrm{cf}}(0)$ (orange) and the curvature sensor change of resistance $\Delta R_c = R_c - R_{\mathrm{cf}}(0)$ (blue), the middle portion shows the tracking error $R_{\mathrm{c}}^{ref} - R_{\mathrm{c}}$ (red), and the bottom section shows the control input for the set pressure (dark green) and the measured pressure (light green). Gravity in all scenarios is acting down. **(A)-(C)** the pressure used to compose the reference signal is within the working range of the finger. **(A)** The finger is curling down, **(B)** the finger is curling to the side, and **(C)** the finger is curling up. **(D)** The anti-windup keeps the finger from becoming unstable when the reference resistance exceeds the working range saturating the pressure; the finger is curling down.

5.3.2 Feedback Control Enabled Touch Detection

After the validation of our controller D_c with the triangular signal (Fig. 6) and freely moving finger, we experimented with a sinusoidal reference signal with an idea to smooth out the transition between positive and negative reference slopes which are associated with the finger flexion and extension, respectively. From this, we found that the magnitude of the reference tracking error is not necessarily smaller, but more evenly distributed in time (see Figs. 6A and 7B). We found that this is beneficial for our method of touch detection, therefore, the experiments we present here are done with the sinusoidal reference signal with a period of $4s$.

In the experiments presented here, the finger base was perpendicular to the surface with the object and mounted $0.057m$ above the top of the surface. The box had the dimensions of $0.03m \times 0.03m \times 0.03m$. The height between the tip of the finger and the surface was $0.0358m$, and the horizontal distance between the tip of the finger and the front edge of the box was $0.0174m$.

The reference tracking controller puts the finger in motion, tracking a sinusoidal reference, flexing the finger, and increasing the curvature sensor resistance until it has come in contact with an object. The blocked movement causes the tracking error between the reference signal, R_{c}^{ref}, and curvature sensor resistance, R_{c}, to increase. Once the threshold, T, is reached and

$$\left| R_{\mathrm{c}}^{ref} - R_{\mathrm{c}} \right| > T, \tag{5.9}$$

contact with the object is maintained by switching the reference to a constant curvature, which allows the finger to maintain its contact. The threshold is chosen to be a value slightly higher

than any error value seen while the finger is in free displacement (shown in Fig. 7B).

Once the the threshold had been reached, we experimented with two reference switching methods. The first one resulted in a permanent, but *lighter touch* between the finger and the box (Fig. 7C), and the other one resulted in a *firm touch* between the finger and the box (Fig. 7E).

We show in Fig. 7C an example of maintaining a lighter touch, which is achieved by the reference signal dropping to the value achieved by the curvature sensor when the touch has been detected (shown in Fig. 7D). If the value ΔR_T corresponds to the value ΔR at the discrete time step k_T at which the threshold is reached, which is marked by the vertical dotted light blue line in Fig. 7D, then the reference of the digital controller ΔR_c^{ref} is substituted with $\Delta R_c^{ref}(k) = a\Delta R_c^{ref}(k-1) + (1-a)\Delta R_T$, i.e., with

$$R_c^{ref}(k) = aR_c^{ref}(k-1) + (1-a)(\Delta R_T + R_{\text{cf}}(0)), \tag{5.10}$$

for all subsequent discrete time steps $k > k_T$, where $R_{\text{cf}}(0)$ the curvature sensor resistance for $p = 0$, $a = e^{-\lambda h}$, h is the sample time of the digital controller and λ is a constant of the exponential transition. This exponential transition prevents a sharp change in the reference signal to the new value. This smooth transition allows for fewer oscillations as the curvature controller adjusts to tracking the new constant level ΔR_T of the reference signal.

In Fig. 7E we present the data from the case when a firm touch is achieved by the sinusoidal reference signal switching to maintain a constant value given by the value of the reference signal at the time of touch detection (shown in Fig. 7F). In other words, once the threshold is reached at the discrete time step k_T, which is marked by the vertical dotted light blue

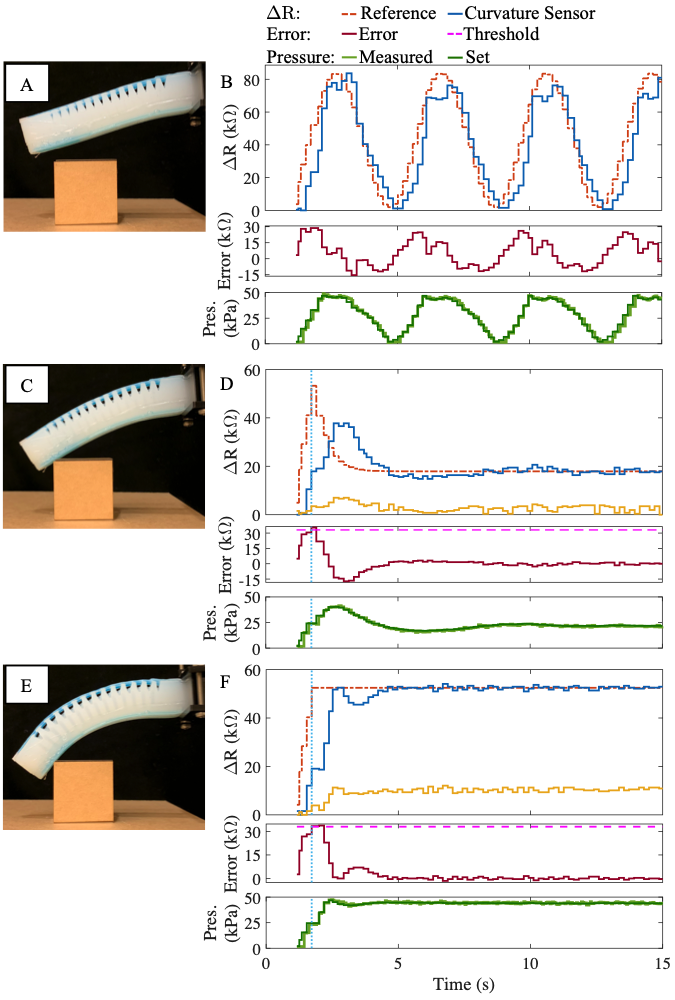

Figure 5.7: **The threshold based switching for the curvature tracking controller when detecting a touch.** **(A)** The finger is positioned above the box prior to beginning a flexing motion. **(B)** The reference signal $\Delta R_c^{ref} = R_c^{ref} - R_{cf}(0)$ is a sinusoidal signal while the finger is unimpeded in free displacement used to determine the threshold. **(C)** The finger after the contact is detected and the reference signal ΔR_c^{ref} is reduced to a value to maintain a light touch. **(D)** The motion is impeded at $t \approx 1.54s$, the touch is detected at $t = 1.72s$ and the finger maintains the light touch on the object ($t > 4.65s$). **(E)** The finger after the contact is detected and the reference signal ΔR_c^{ref} is held at a value to maintain a firmer touch. **(F)** The motion is impeded at $t \approx 1.54s$, the touch is detected at $t = 1.72s$ and the finger maintains the firmer touch on the object ($t > 4.23s$). In all data plots, the reference signal is plotted as $R_c^{ref} - R_{cf}(0)$ (orange), the finger curvature sensor resistance is plotted as $\Delta R_c = R_c - R_{cf}(0)$ (blue), force sensor (yellow), the tracking error $R_c^{ref} - R_c$ (red), the threshold (pink) at $T = 33k\Omega$, the control variable, i.e., the set pressure (dark green), and the measured pressure (light green) are shown.

line in Fig. 7F, then for all time steps $k > k_T$, $\Delta R_c^{ref}(k) = \Delta R_c^{ref}(k_T)$, i. e.,

$$R_c^{ref}(k) = R_c^{ref}(k_T). \tag{5.11}$$

This creates a firmer touch since the higher reference value requires a higher pressure.

5.3.3 Summary

The experiment results show that we were able to successfully close the feedback loop, in which the curvature sensor resistance follows the reference signal. The feedback control design is based on information about the bounded slope of curvature sensor characteristics with respect to the inflation pressure and the requirement that the reference signal is a triangular signal. However, through experimentation we found that a sinusoidal reference signal resulted in a reference error which changes more evenly with the change of the reference.

Our results prove the concept of touch detection which uses only the curvature sensor resistance. To achieve that, we did not use any relation between the curvature sensor resistance and the soft finger bending. All we needed is to establish the feedback loop for the curvature sensor resistance to track a reference signal. The touch detection is based on the level of the tracking error. Once a threshold error is reached, the reference changes to maintain the touch. We found that the touch can be maintained with two types of reference, one which maintains a light and the other which maintains a firm touch.

All the presented results were achieved with the real soft finger and the system that included various imperfections. As shown in Fig. S1-S2, for example, there was both noise and some delay induced by the digital regulator, which functioned by opening and quickly closing a

valve to let in air from a high-pressure source.

5.4 Conclusion

This paper presented and validated the feedback control enabled touch detection for the soft finger. As part of the validation, we also described the soft finger and the validation system used in our experiments. With this, we have shown that the presented method does not ask for special soft finger design requirements, and that method can be applied to a large number of finger designs that include a curvature sensor. Without using a force sensor, our method removes the requirement for the collocation between the force sensor and point of touch.

The overall method does not depend on any external sensing technologies such as a motion capture system which may not be present in real-world applications. It is expected that the proposed touch detection method can be used with other sensing modalities which can be embedded in a soft finger for further improvements of soft robotic grippers and hands based on proprioceptive feedback.

Chapter 6

Compliant Proprioceptive Touch without a Force Sensor: A Kinesthetic Feedback Control Approach

This chapter is a preprint to the paper

- M. Boivin, D. Milutinović, and M. Wehner, "Compliant Proprioceptive Touch without a Force Sensor: A Kinesthetic Feedback Control Approach," (in prep).

6.1 Introduction

While compliant robotic manipulators have significantly improved over the last few years [110], there are still challenges in their operation related to their interactions with objects of unknown location, shape, orientation and surface characteristics. To address these challenges, many manipulators rely on processing visual data from cameras. However, this source of data

can be unreliable in dealing with transparent objects [57, 107], a reduced light environment, or blocked camera view by a manipulator approaching an object to grasp. In these situations, humans don't only rely on visual feedback, but also on the sense of touch through proprioceptive and tactile sensory systems of their fingers. That and recently developed compliant soft robotic fingers [118] with embedded sensors have inspired our work on the feedback control enabled sense of touch.

In humans, the sense of touch is described by two modalities, *cutaneous* and *kinesthetic* [27]. The cutaneous sense of touch refers to the modality in which "the sensory input is from the receptors embedded in the skin, and the kinesthetic sense receives sensory inputs from the receptors within muscles, tendons, and joints" [27]. When creating an analogy for robots, the cutaneous sense corresponds to the sense of touch provided by dedicated sensors of touch [11, 12, 27, 52, 126] and the kinesthetic sense of touch is the one derived from a tension or stress from actuation, which is the focus of our paper.

Previously, kinesthetic information has been used in haptic displays [10], including those with soft-robotic components [60] and other applications with the sense of touch [48], in which the contact sensor information was combined with its position. However, up to our best knowledge, information about a finger-like actuator position and an effort to control it as measured by a tracking error of feedback control loop has not been previously considered to detect and maintain a contact between the actuator and an object in its operating space.

We propose here a reference tracking feedback control of a compliant finger, which uses the finger's curvature measurements and enables the sense of touch. The tracking control allows to set the finger either to a desired curvature value, or to follow a desired curvature

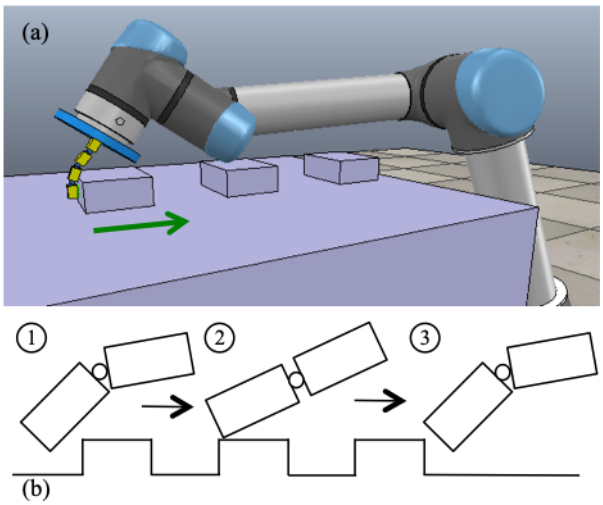

Figure 6.1: The compliant finger in a V-REP scene: (a) the finger mounted on the UR5 arm and in a firm touch with the first box; (b) the finger reacting as it is dragged across the objects. The curvature is reduced as it comes into contact with an object and returns to the reference curvature once it is no longer in contact.

reference signal, which results in a controllable motion of the finger over configurations in the range between full flexion and extension. The tracking error of the control loop is minimal when the finger's motion is unimpeded, i.e., not in touch with any object. When the finger is in touch with an object, the error increases and its intensity measures the amount of stress on the finger.

Fig. 6.1 (top panel) shows a virtual robot experimentation platform [104] (V-REP) scene with a compliant finger which is created to replicate fundamental properties of the soft finger [118], its actuation and curvature measurement. In the figure, the finger is in a controlled firm touch with an object. If the finger is dragged, the tracking error can be used to detect objects, their edges (see Fig. 6.1, bottom panel), and the level of the error can be used as an indication of object heights. Controlled motion of the finger also allows for tapping objects and visualizing objects and their textures following similar principles that are used in atomic force microscopy [46], but at a scale of the finger's size. All these capabilities of a single finger can be

further exploited on a multi-finger robotic gripper and used as a building block for its controlled grasp. Our work is a significant step towards control principles and mode of uses of such soft robotic grippers.

Soft robotic components show promising results in applications involving grasping [110], biomedical applications [25], and wearables [97,100] through various forms of locomotion and sensing [67]. They belong to the larger class of soft compliant robotic components with flexibility that allows them to adapt to their operating environment and handle higher levels of uncertainty in object manipulation than traditional rigid robot components. These include fingers with rigid links and flexible joints [94], as well as soft material robotic components [62, 82, 118], grippers [29, 55, 56, 80, 89], and multi-segment manipulators [81, 83, 85].

Due to their flexibility, compliant robotic components have less constrained movements and are difficult to control [83], especially when the component is in contact with objects in the operating environment [55]. Consequently, both the model-based and machine-learning-based control methods have been used to control soft robotic components.

A variety of soft robotic actuators have been developed over the past decade ranging in design, sensors, modeling, and control strategies [98, 102, 105, 120, 122], which are also surveyed in [42]. The model-based control methods are based on a piece-wise constant curvature kinematics model [32,61,124] and the use of system identification technique [85] to estimate the piece-wise constant curvature model parameters. The machine-learning based methods include the use of neural networks [20,114] and reinforcement learning [113]. In addition to these, some authors also considered a combination of model-based and machine-learning methods [116].

Feedback control has been used in various applications and has been based on camera

system visual feedback [62, 81, 83, 85], or dedicated sensor measurements [55, 56, 89]. Aside from a practical application showing that feedback control for soft systems is possible, in [30], the authors showed that higher feedback gains lead to higher perceived stiffness of an inherently soft robotic component. This is a significant result for systems that rely on the compliance property of the robotic component as in the work presented here.

Our approach to detect and maintain contact between the compliant finger and an object is inspired by kinematic problems [86], in which the anticipation of future relative motion between system variables is instrumental for robust decision making. The feedback control that we propose employs the control to set the compliant robotic finger curvature to a certain position, or reference movement, which serves as a *prior* for the movement. The approach uses the discrepancy between that movement and the actual one to infer robustly that the finger is in contact with the object, so that the edge can be detected, see Fig. 6.3.

In our preliminary study for the control of a soft robotic finger [13], we used a continuous-time second order controller along with a deterministic static characteristic relation between the inflation pressure and curvature. Here we use system identification to estimate a dynamic model of the finger. To achieve that, we develop a procedure on how to avoid overfitting the data, from which we learn that the finger can be represented and successfully controlled by a second order discrete-time domain linear model. We use the identified model to design a reference tracking digital controller. However, in order to use the feedback controlled finger for the sense of touch, we have to evaluate the controller performance for the finger that is mounted on a robot arm. This provides us tracking error distributions for various orientations and motions of the robot arm, and thresholds that we use for edge detection, or firm touch by the finger.

The paper is organized as follows. Section 6.2 formulates the problem of edge detection utilizing proprioceptive feedback. A system identification approach for modeling the curvature of the finger is described in Section 6.3. The control architecture is described in Section 6.4, with the reference tracking edge detection defined in Section 6.4.4. Section 6.5 presents results from V-REP simulations in which the finger is used to detect object edges and visualize objects. Section 6.6 gives conclusions.

6.2 Problem formulation

Figure 1 shows a V-REP scene with the soft finger mounted on a robot arm that we use in our study. The complexity of soft finger dynamics is replicated in V-REP by a series of four-rigid body links connected with three compliant elastic joints. The single input control variable for the finger is a torque, which is distributed over the joints. This torque actuation of the V-REP finger emulates the inflation pressure that actuates the real soft finger [118], therefore, we denote this control variable by p.

When the control input is $p = 0$, the finger configuration depends on a relative orientation between the joint axes and gravitation acceleration. Figure 2*a* shows that the finger naturally bends under the influence of gravitation when the input $p = 0$. When the input is $p > 0$, the finger bends further (see Fig. 2*b*). Both of these outcomes match characteristics of the soft finger [118].

In the real soft finger [118], the degree of finger bending is measured by a curvature sensor. The sensor is implemented by a resistor which is mounted along the finger and changes

its resistance with the finger bending. In our V-REP scene model, we ”measure” the curvature ΔR_{cf} as

$$\begin{aligned} \Delta R_{cf} &= K \sum_{i=1}^{3} s_i \\ s_i &= \begin{cases} \theta_i (r_i + d) & \text{if } \theta_i \neq 0, \\ 2l & \text{else} \end{cases} \end{aligned} \tag{6.1}$$

where K is a scaling factor, θ_i is the joint i position angle, which is the angle between $\overrightarrow{O_i B_i}$ and $\overrightarrow{O_i A_{i+1}}$ (see Fig. 6.2b), d is the offset from the joint center to the top of the fragment, $2l$ is the distance $\overline{B_i A_{i+1}}$ for $\theta_i = 0$, and r_i is the distance between the point O_i and the corresponding ith joint computed as $r_i = l(\tan(\theta_i/2))^{-1}$. In expression (6.2), the variable s_i is the length difference between the arc from B_i to A_{i+1} for $\theta_i \neq 0$ and $2l$.

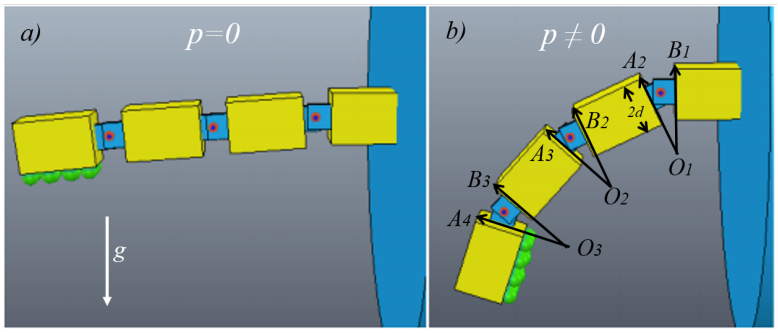

Figure 6.2: The complaint finger in V-REP: a) shows the finger under the effect of gravity with no pressure applied; and b) shows how the finger bends when the pressure is applied. The arc lengths across the joints are computed to “measure” the bending of the finger.

Figure 6.3 shows the block diagram of the feedback controlled finger, where the controller $D_c(s)$ computes the control variable p from the difference between the curvature measurement ΔR_{cf} and the curvature reference signal ΔR_{cf}^{ref}. If the controller $D_c(s)$ provides the zero error tracking of the curvature reference, then an obstacle that impedes the finger motion

results in a non-zero tracking error signal. Since the finger is flexible, this signal can be used as a measure of the contact force between the finger and the obstacle.

Two problems that we consider here are (P1) the design of a feedback control for the finger curvature reference tracking, and (P2) the formulation of the reference curvature signal for specific applications.

To address *P1*, we perform a system identification of the finger dynamics and use the identified model for the control design. For *P2*, we demonstrate the controller's work in multiple scenarios.

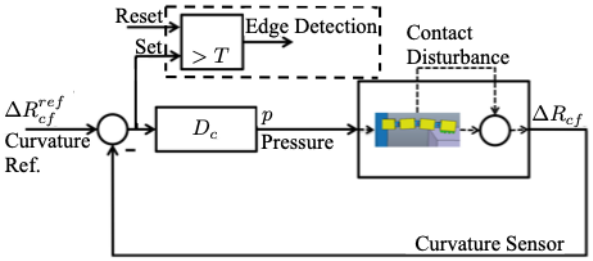

Figure 6.3: The feedback control loop. The feedback controller D_c input is the tracking error, which is the difference between the reference and measured curvature of the finger, and the controller output is the pressure. For an unimpeded motion of the finger, the tracking error is small. If the error passes a threshold, it indicates that the finger is in contact with an object. Before the movement, the block that implements the detection threshold is reset to 0.

6.3 System Identification

For the model of the finger dynamics, we use the linear ARX discrete time model [73]

$$y(z) = G(z)p(z) + H(z)e(z) = \frac{B(z)}{A(z)}p(z) + \frac{1}{A(z)}e(z), \tag{6.2}$$

where $p(z), y(z)$ and $e(z)$ are z-transform images of the discrete time control variable $p(t_k)$ and the curvature measurement variable $y(t_k) = \Delta R_{cf}(t_k)$, and $e(z)$ is the z-transform of the model

error signal $e(t_k)$, respectively, and $k = 1, 2, ...$ The model sample time is $t_{k+1} - t_k = 0.05$ seconds, and $B(z)$ and $A(z)$ are polynomials of the transfer function $G(z)$ defined as

$$A(z) = 1 + a_1 z^{-1} + \cdots + a_n z^{-n} \tag{6.3}$$

$$B(z) = b_1 z^{-1} + \cdots + b_m z^{-m}, \tag{6.4}$$

where $n \geq m \geq 1$. In the time domain, this model results in the linear difference equation

$$y(t_k) + a_1 y(t_{k-1}) + \cdots + a_n y(t_{k-n}) = b_1 p(t_{k-1}) + \ldots$$
$$\cdots + b_{m+1} p(t_{k-m+1}) + b_m p(t_{k-m}) + e(t_k), \tag{6.5}$$

which describes the relationship between the input $p(t_k)$, output $y(t_k)$ and the model error $e(t_k)$, $k = 1, 2, ...$ For this model, we estimate the vector of unknown coefficients

$$\theta = \begin{bmatrix} a_1 & \ldots & a_n & b_1 & \ldots & b_m \end{bmatrix}^T, \tag{6.6}$$

as well as their number, i.e., orders of the transfer function polynomials $A(z)$ and $B(z)$. We estimate them based on the assumption that the sequence of model errors $e(t_k)$ is a zero-mean white noise.

Procedure: We identify the finger dynamics from an open-loop experiment, in which the input $p(t_k)$ is a biased pseudo random binary sequence defined as

$$p(t_k) = p_{bias} + u_b(t_k),\ p_{bias} = 1 \tag{6.7}$$

$$u_b(t_k) = \begin{cases} -u_b(t_{k-1}) & \text{with probability } \pi = 0.5 \\ u_b(t_{k-1}) & \text{with probability } \pi = 1 - \pi \end{cases}$$
$$u(0) = \{-1, 1\}.$$

The bias p_{bias} is added to shift the range of the pseudo random input $p(t_k)$ to be only positive since the finger cannot bend backward.

For the sequence $p(t_k)$, we collect the output data $y(t_k)$ and form the vector

$$\phi(t_k) = \begin{bmatrix} -y(t_{k-1}) & \dots & -y(t_{k-n}) & u(t_{k-1}) & \dots & u(t_{k-m}) \end{bmatrix}^T, \tag{6.8}$$

which can be used to express a one step prediction of the output $\hat{y}(t_k) = \phi(t_k)^T\theta$, for given θ. Based on this, we estimate the parameter vector θ by minimizing the sum of squares of one step prediction errors for t_k, $k = n+1,...N$ and $N >> n$ as

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N-n+1} \sum_{k=n+1}^{N} (y(t_k) - \phi^T(t_k)\theta))^2. \tag{6.9}$$

The minimization of the prediction errors results in the solution

$$\hat{\theta} = \left[\sum_{k=n+1}^{N} \phi(t_k)\phi^T(t_k) \right]^{-1} \sum_{k=n+1}^{N} \phi(t_k)y(t_k), \tag{6.10}$$

where the right side of the solution is completely defined based on input and output data recorded at t_k, $k = 1,...N$.

Once we have recorded the data, we compute the estimate $\hat{\theta}$ for an increasing sequence of the polynomial $A(z)$ order $n = 1,2,\dots$ and using $m = n$ for the order of polynomial $B(z)$. We increase n until we do not see more than 1% in the reduction of the sum of squares of prediction errors, which we use as a criterion to prevent the overfitting of the data. From this, we obtain the 6th order model with parameters resulting in

$$B(z) = 1.419z^{-1} + 0.3242z^{-2} - 0.1622z^{-3} - 0.5952z^{-4} - 0.443z^{-5} - 0.5036z^{-6} \tag{6.11}$$

and

$$A(z) = 1 - 0.2956z^{-1} - 0.2534z^{-2} - 0.2808z^{-3} \\ - 0.1212z^{-4} - 0.0847z^{-5} + 0.0371z^{-6}, \quad (6.12)$$

which is the highest order model that we use to describe input-output relations from our data. The model has six poles and five zeros (see Fig.6.4) of which the pole at 0.9994+j0 and the zero at 0.9937+j0 are almost on the top of each other. The other poles also show up in pairs with zeros in their proximity. Pairs of poles and zeros of a transfer function that are in close proximity of each other can be canceled. Therefore, the presence of such pairs in our model indicates the possibility of overfitting, i.e., that our model is of a higher order than necessary. To understand the impact of pole-zero pairs, we proceed with the analysis of the model in the frequency domain.

Model analysis through model reduction: The analysis consists of a comparison of the original transfer function $G(z) = \frac{B(z)}{A(z)}$ with the simpler model $G'(z)$ obtained from $G(z)$ by removing one or more pole-zero pairs, while keeping the DC gain unchanged, i.e., $G'(1) = G(1)$. As an example, Fig. 6.5 shows the Bode plot of $G(z)$ and $G'(z)$, which is obtained by the removal of the pole and zero in the proximity of $1 + j0$. In this specific case, it is clear that $G'(z)$ is a poor approximation of the original $G(z)$. When we sequentially remove pole-zero pairs, we obtain that only in one case $G'(z)$ does not follow the phase of $G(z)$, while in all other cases approximations approximated $G(z)$ quite well both in magnitude and phase until 0.1 rad/sec. Out of these good approximations, the transfer function $G'(z)$ with the smallest order of polynomial is the one with

$n = 2$, which is

$$G'(z) = 1.9502 \frac{z - 0.9937}{z^2 - 1.263z + 0.2632}. \tag{6.13}$$

The poles and zeros of the transfer function are highlighted in Fig. 6.4.

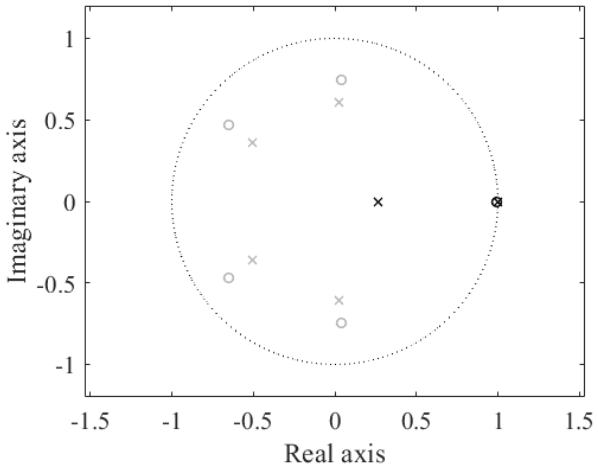

Figure 6.4: Poles and zeros of the discrete models: the 6th order ARX model poles ('x') and 5 zeros ('o'). The poles and zeros of the 2nd order model are indicated with the symbols in bold.

For an illustration, we compute the one step prediction for the 2nd order model to show that for the same input sequence, the model output closely follows the data from our experiments.

6.4 Control Architecture

The controller in Fig. 6.3 is a curvature reference tracking controller. The controller provides that the curvature of the finger ΔR_{cf} (see (6.2)) follows the reference ΔR_{cf}^{Ref}. If the finger is able to move freely, i.e., there is no object in its path, we expect a small tracking error $(\Delta R_c f^{Ref} - \Delta R_c f)$. Therefore, the tracking error signal can be also used as a signal to detect

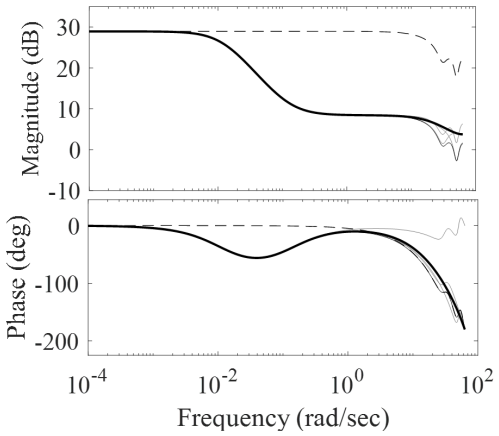

Figure 6.5: The frequency responses. The responses are obtained from the 6th order systems and the removal of combinations of closest pole-zero pairs. The dashed line depicts the frequency characteristics of $G'(z)$, which is obtained by the removal of the pole-zero pair in the proximity of 1+j0.

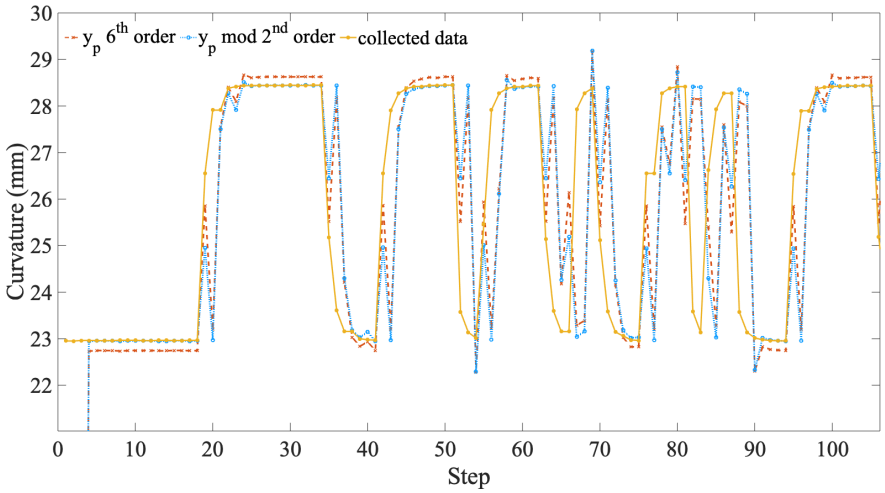

Figure 6.6: The original 6^{th} order ARX and modified 2^{nd} order models and measured outputs for the same input data.

contact of the finger with an object in the robot's workspace. A ramp reference signal sets the freely moving finger in a motion such that it resembles the motion of a finger sampling points in space to touch. We consider this type of motion since it is an integral part of a finger motion during the opening and closing of multi-finger grippers.

6.4.1 Curvature Reference Tracking Control

For tracking the ramp reference signal by the system modeled by $G'(z)$ we use a unity feedback loop with the controller $D_c(z)$. From the discrete linear system control theory, we know that the steady state of the error $e_{ss} = \lim_{t\to\infty} e(t)$ for tracking the reference is

$$e_{ss} = \lim_{z\to 1}(z-1)\frac{1}{1+D_c(z)G'(z)}\Delta R_{cf}^{ref}(z), \tag{6.14}$$

therefore, for the zero error tracking of a unit ramp signal reference that has the z-transform $hz/(z-1)$, where h is the sample time, we need $\lim_{z\to 1}(z-1)D_c(z)G'(z) = \infty$. This can be achieved with the controller that has the form

$$D_c(z) = K_c\frac{z-z_c}{(z-1)^2}, \tag{6.15}$$

where K_c is the controller gain and z_c is the zero that is introduced by the controller. For the loop gain $D_c(z)G'(z)$, we use the discrete root locus analysis for placing the zero z_c and the gain K_c. To tune these parameters, we use criteria of the damping ratio of $\zeta = 0.707$ and the settling time $t_s < 1.5sec$ on the root locus diagram. From these, we tune $K_c = 3.8$ and $z_c = 0.7$. Figure 6.7 shows a numerical simulation in MATLAB of the curvature controller with the determined parameters in free displacement. The range for the reference signal was chosen based on the data that we sampled to identify the transfer function $G'(z)$. This numerical simulation also provides

a baseline threshold value for determining when the finger is in contact with objects or when it stops being in contact with them.

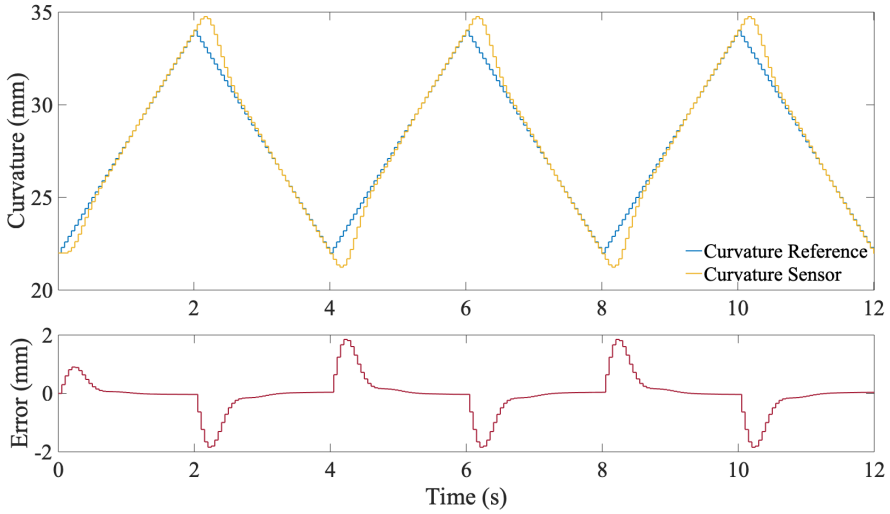

Figure 6.7: The discrete-time curvature tracking controller: (top panel) the reference input (blue) is the triangular signal that results in the flexing and extending of the finger; the curvature sensor measurements (yellow) track the reference input and overlap with it; (bottom panel) the tracking error.

6.4.2 Tracking Error Performance

We implemented the feedback controller in V-REP with the compliant finger mounted on the UR5 robot. Due to the finger's weight and its flexibility, the robot posture and motion could impact the tracking error. Therefore, we ran experiments with the finger in free displacement to find an error threshold that worked well for our system. To collect data about the interference of the motion and robot pose on the unimpeded finger, we performed a total of 20 experiments. These experiments can be broken into three types:

1. The curvature of the finger is maintained at a constant value while the arm moves in an upward motion. Six experiments are executed: the finger is either in a vertical (perpen-

dicular to the ground) or horizontal (parallel to the ground as seen in Figures 6.10a_1-a_6) orientation to begin and for each orientation, it curls down toward the ground (Fig. 6.10a_4), to the side (Fig. 6.10a_5), and up toward the ceiling (Fig. 6.10a_6). The vertical orientation can be seen in Figures 6.8a_1-a_6 and the data from the experiments seen in Figures 6.9a_1-a_2 shown in the appendix.

2. The finger flexes and extends between the curvature range for roughly a four- and a two-second period between extensions, respectively, while the arm moves in an upward motion. Six experiments are executed for each period for twelve total experiments: the finger is either in a vertical (perpendicular to the ground) or horizontal (parallel to the ground as seen in Figures 6.10a_1-a_6) orientation to begin and for each orientation, it curls down toward the ground (Fig. 6.10a_4), to the side (Fig. 6.10a_5), and up toward the ceiling (Fig. 6.10a_6). The vertical orientation can be seen in Figures 6.8a_1-a_6 and the data from the experiments seen in Figures 6.9b_1-c_2 shown in the appendix.

3. The finger flexes and extends between the curvature range for roughly a four- and a two-second period between extensions, respectively, while the wrist rotates $0^\circ - 360^\circ$. Only the horizontal orientation for the two different periods was considered (shown in appendix Figures 6.8c_1 and c_2 and data shown in Fig. 6.9d).

Figure 6.10a and Figure 6.8 in the appendix shows the setup for these experiments, where the finger tracks a reference input while the arm moves upward or the finger base rotates in a circle. In the first set of experiments listed (appendix Fig. 6.9a), the curvature controller tracks a ramp signal in order to maintain a smooth motion of the finger until the desired curvature has

been reached, at which point it holds steady. In the other experiments (appendix Fig. 6.9b-d), the controller tracks triangular signals composed of ramp inputs with periods of two and four seconds. Through these experiments, we can see how the residual error of the system is affected by different types of motion of the finger. From the data, we have concluded that the error is largely due to the type of reference signal used, and the motion of the arm has little effect on the error. Collecting the tracking errors from the experiments, we can use the distribution of these errors (Fig. 6.10b-e) to find thresholds that indicate an impeded motion. If the error of the system is below the threshold, then it is an indication that the finger is in free displacement and has not made contact with an object in its path.

From the first set of experiments (Fig. 6.9a), we can see that once the finger reaches the steady state of the constant curvature reference, at about $t = 1s$, the motion of the arm alone does not affect much the error of the system, which has an error very close to zero. The error distribution across the six experiments for $t > 1s$ is shown in Fig. 6.10b. However, in the experiments where the finger tracks a reference composed of ramp signals (appendix Figures 6.9b-d), the error is an order of magnitude larger, and decreasing the period (i.e., increasing the speed of the flexion and extension of the finger) increases the error even more. The error distribution of the arm in motion for a triangular reference of four and two seconds is shown in Figures 6.10c and d, respectively. Figure 6.10e shows the combined finger base rotation experiments for both two and four seconds. For our purposes, a reference that causes the finger to do a full flexion and extension in roughly 2 seconds is sufficient. Since the error for a constant curvature reference compared to a ramp reference varies by an order of magnitude, we chose to use two thresholds. To choose the threshold, we looked at the distributions across all experiments as

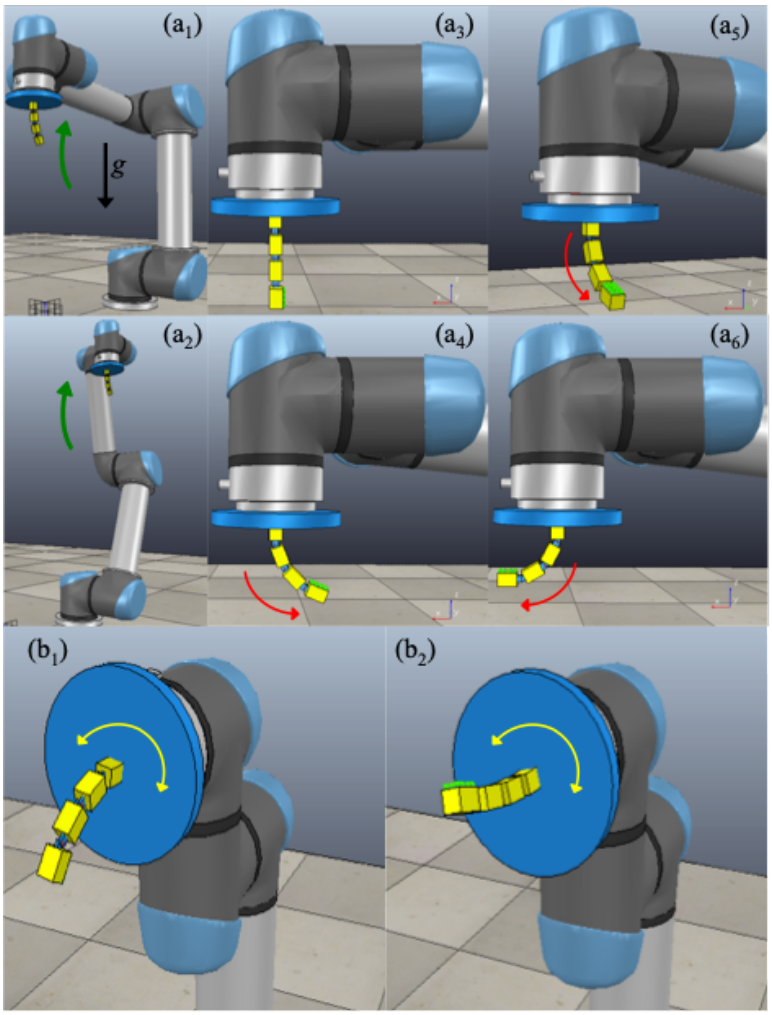

Figure 6.8: (a_1-a_2) Shows the arm in the starting and end positions with the finger in the vertical orientation. The green arrow indicates the direction the arm moves to reach the final position. The finger starts off straight and either curls to a given static reference curvature or tracks a periodic ramp signal flexing and extending while the arm moves upward. Direction of the finger's motion is indicated by the red arrorw. (b_1-b_2) The finger is in the horizontal orientation and tracks two different ramp signals flexing and extending while the wrist of the arm rotated from 0° to 360° and back. The yellow arrow indicates the direction the wrist rotates.

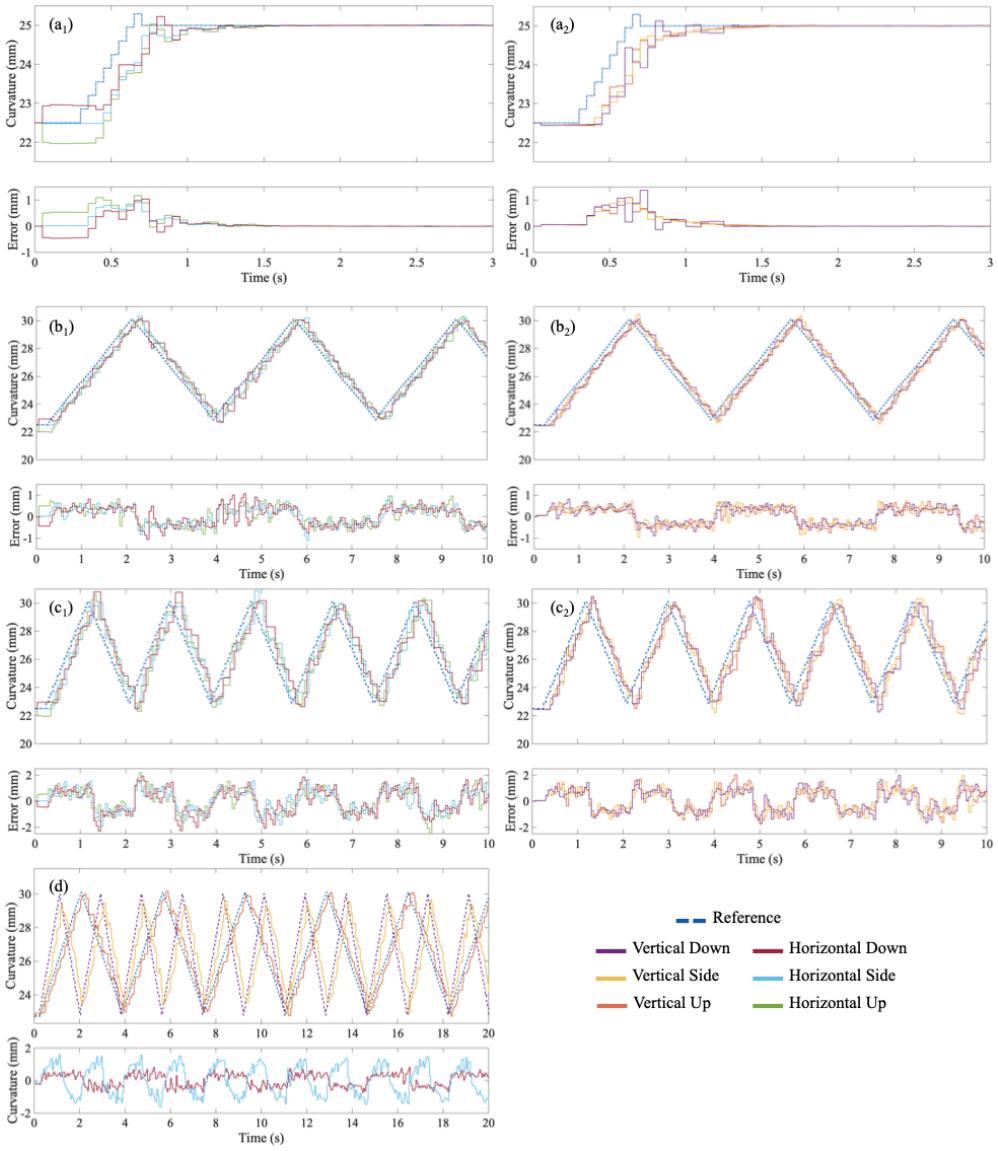

Figure 6.9: (a_1-c_2) Data was collected for three positions of the finger tracking the respective reference curvature (top panel): curled downward, curled to the side, and curled upward, along with the residual error (bottom panel). The finger was held at a static reference curvature for the finger in (a_1) the horizontal orientation and (a_2) in the vertical orientation. The finger tracked about a 4-second periodic ramp signal flexing and extending while the arm moved upward for the finger in (b_1) the horizontal orientation and (b_2) the vertical orientation. The finger tracked about a 4-second periodic ramp signal flexing and extending while the arm moved upward again for all three positions of the finger (c_1) in the horizontal orientation and (c_2) the vertical orientation. (d) Data collected for a 4- and a 2-second periodic ramp signal flexing and extending while the wrist of the arm rotated from 0° to 360° and back.

shown in Fig.6.10. For a constant curvature reference, we chose a threshold $T = 1$. Based on the distribution of errors, the threshold can be smaller, but with this value, we account for a required control effort to return the curvature to its reference value after the finger is no longer in contact with an object. For a ramp reference, we chose a threshold of $T = 3$ to be slightly larger than the max error seen at $|e_{max}| = 2.5$. A look up table can be used to apply the threshold that corresponds to a desired type of the reference signal. Since there are limits on the amount of torque the finger can withstand, the control architecture included anti-windup to stop the integration and accumulation of the error when reaching the saturation limits [9], which is also determined by these thresholds.

6.4.3 Firm Touch

Figure 6.11 shows the experiment in which the finger moves and stops once it touches an obstacle. The controller puts the finger in motion tracking a ramp reference, flexing the actuator, and increasing the curvature until it has come into contact with an object. The blocked movement causes the residual error between the reference signal and curvature sensor to increase. Once the threshold, T, is reached and

$$\left|\Delta R_{cf}^{ref} - \Delta R_{cf}\right| > T, \tag{6.16}$$

a firm touch with the object is maintained by switching the reference to a constant curvature, which allows the finger to maintain its contact.

In this experiment, the finger base is rotated 60° from the x-axis of the world frame and held $0.05m$ above the top of the box. The box has the dimensions of $0.10m$ *x* $0.10m$ *x* $0.080m$

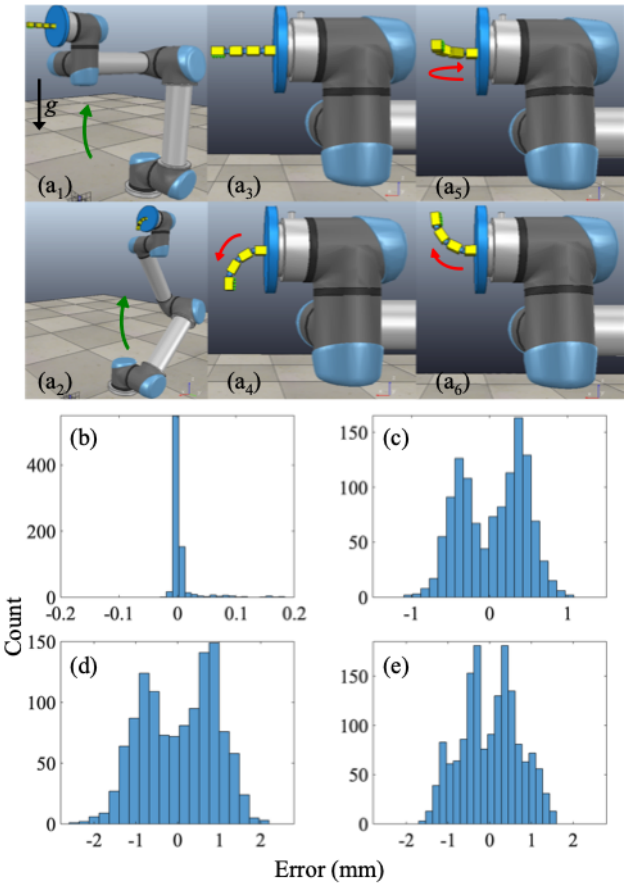

Figure 6.10: Tracking error distributions: (a) The finger in the horizontal orientation. The green arrow shows the direction of the arm's motion, and the red arrow indicates the direction the finger curls. (a_1) The initial position of the arm. (a_2) The final position of the arm. (a_3) The starting position of the finger without any torque. (a_4) The finger curls downward. (a_5) The finger curls to the side. (a_6) The finger curls upward. Each histogram is obtained from errors across all the experiments that use the same type of the reference signal or the robot arm wrist motion: (b) Constant curvature reference. The errors were collected once the finger reached equilibrium and settled around the reference signal. (c) A 4-second periodic triangular reference signal. (d) A 2-second periodic triangular reference signal. (e) Experiments where the wrist rotates for both 4- and 2-second periodic triangular reference signals.

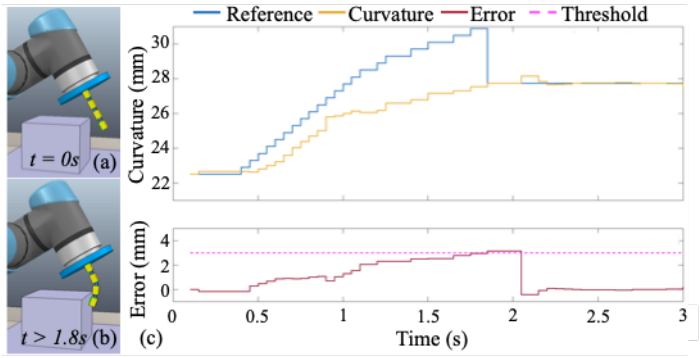

Figure 6.11: Firm touch: (a) The finger is positioned in front of the box, and begins a flexing motion. (b) The motion is impeded at $t \approx 0.7s$, and the finger maintains a firm touch on the object ($t > 1.8s$). (c) The finger curvature (yellow), the ramp reference signal (blue), the tracking error (red), the threshold (pink) at $T = 3mm$.

and weighs $0.5kg$. The arm is stationary, and we used the curvature controller from the previous section to actuate the finger using a ramp signal with a period of two seconds. From Fig. 6.11c, we can see that initially the finger follows the ramp reference, but then it comes into contact with the object (Fig. 6.11a) and while it continues to flex, the rate of flexing does not match the reference signal rate, which can be noticed in the increase of the tracking error (Fig. 6.11c). Once the error crosses the threshold, the reference switches to a curvature value required to maintain a firm touch and the controller maintains that curvature once reached. The finger remains in firm contact with the object as depicted in Fig. 6.11b.

6.4.4 Edge Detection

Extending the idea of movement error based control even further from a firm touch on an object, we can utilize the same concept of crossing a threshold for detection of edges. An edge of a surface is detected when there is a transition in the error from the reference tracking controller crossing a threshold. While the finger is in contact with the surface, the error will

be higher than the predetermined threshold since the finger is impeded. As the finger is moved across the surface, when it leaves the surface, the error will decrease back to zero since the finger is no longer blocked and able to resume tracking the original reference signal. The feedback control loop and the edge detection are depicted in Fig.6.3. The edge detection block compares the tracking error to the threshold T and has set and reset inputs. Before activating the curvature controller $D_c(z)$, the output of the block is reset. Assuming the arm moves into a position in front of objects in the workspace, the front edge of an object has been detected if Eq. 6.16 holds true. Once this occurs, the output of the block is set to 1 to indicate an edge has been detected. The signal remains high until another transition is made when

$$\left|\Delta R_{cf}^{ref} - \Delta R_{cf}\right| < T \tag{6.17}$$

detecting that the finger has left the object either by being dragged across the surface or lifted away. The block is then reset to allow for detecting multiple edges.

6.5 Results

We present in this section five simulation experiments that were performed using V-REP. The goal of the experiments was to demonstrate the functionality of the combination of the proposed feedback control, reference signals and the control parameters from the previous section, and integrate with the robotic arm to perform motions and object detection with the sense of touch. In the experiments, we explored two different methods for detecting edges of objects. In both cases, the arm is in motion in either the x or y direction, and the finger is held at a constant height after moving into position in front of the object(s). The first case uses a constant

curvature reference for the finger and the second one the curvature reference is a triangular reference signal.

Constant curvature reference experiments: The workspace for these experiments is depicted in Fig. 6.12$a_1 - a_2$. In these experiments, the finger sweeps the y-direction of the workspace at a speed of roughly $0.02m/s$ without being lifted away. Due to the finger's flexibility, its curvature naturally decreases while being dragged across the three boxes. The transition from a higher to a lower curvature triggers the threshold rule (6.16), indicating the finger has made contact with the front edge of a box. The curvature of the finger is relaxed while the finger remains in contact with the object. Once the finger is dragged off the back edge of the box, there is another transition in the curvature, this time from low back to the reference signal, which triggers the threshold rule (6.17), indicating the finger has left the back edge of the box. In these experiments, we use the threshold, $T = 1mm$.

We first ran an experiment where there were three boxes of varying heights of $0.045m$, $0.035m$, and $0.025m$. The largest and middle boxes were separated by $0.115m$ and the middle and smallest boxes were separated by $0.105m$. The feedback control was used with three different curvature reference values, $23mm$, $24mm$, and $25mm$ to see how varying the amount the finger was initially curled affected the edge detection. In Fig. 6.12$a_1 - a_6$, we show the process of the finger being dragged across the boxes, and the resulting data is given in Fig. 6.12b for the different curvatures. The data show that the different curvatures have a slight offset in the tracking error, as well as a slight variance in the time when the edges were detected. This is due to the different starting curvatures and the anti-windup method used. To prevent the accumulation of error, once the error threshold was reached, the integration was stopped and the curvature of

the finger was allowed to relax. When the finger left the object, the integration resumed using a previous value of the control effort that was needed before the edge was detected and when the error was small, $e(k) < 0.25$. This method was used to allow the finger to remain compliant and bend freely so that it did not become stiff like its rigid counterparts. It is important to note that in this case, we found that due to the finger's compliant nature, the transition of being dragged off an object sometimes caused oscillations that were read as multiple edges when there should have been only one. To handle this, we implemented debouncing to ensure that only a single threshold crossing would be acted upon. To debounce the signal, we checked for four consecutive samples of the edge signal to determine whether the threshold was truly crossed. Without debouncing, the oscillations may cause unpredictable results.

Interestingly, it can be seen that the error signal in Fig. 6.12b also gives some insight into the height difference of the objects seen in Fig. 6.12. This is due to the arm maintaining the same height across all the boxes as they decrease in height. Since the finger hits at different heights on the boxes, the amount that the curvature of the finger is affected varies. Utilizing this feature, we preformed the second experiment in which we created a search pattern that searched the workspace plane in the x and y directions. After each path in the y-direction, the finger moved across the x-direction in increments of $0.005m$, which is half the width of the finger. This broader search pattern was used to create a mapping of the workspace utilizing the tracking error signal to detect multiple objects. The result of that mapping can be seen in Fig. 6.13b. The varying heights are indicated by the change in color which represents the amount of error seen at that point in space. The data also show a fairly accurate reconstruction of the workspace plane, where all the boxes have a width of $0.1m$ and length of $0.075m$.

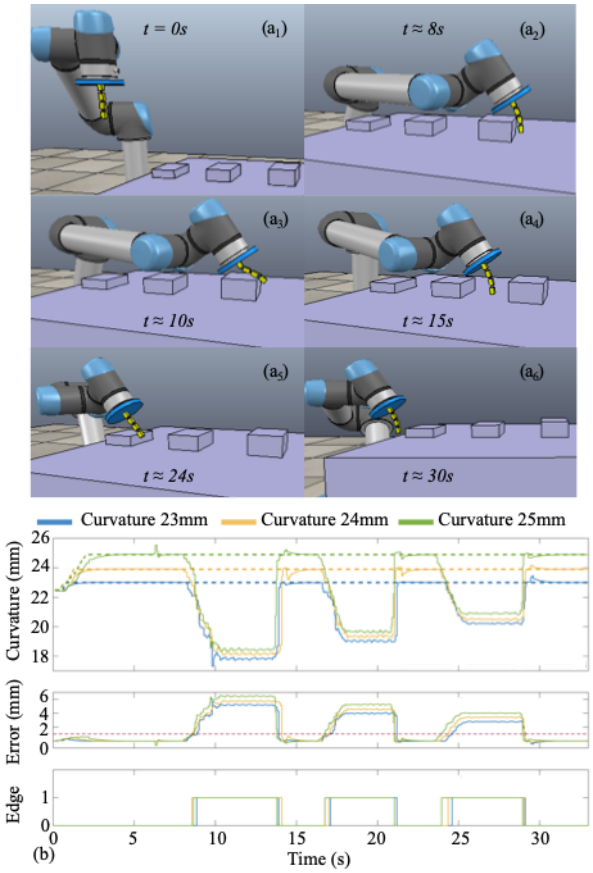

Figure 6.12: Dragging finger scene: (a) The finger is attached to a robotic arm that moves the finger to the front of the first box on the table and drag the finger across the boxes. (b) The curvature, the tracking error, and edge binary signals for three different curvature references, 23*mm* (blue), 24*mm* (yellow) and 25*mm* (green).

A third experiment was done utilizing the same search patter as described above, but this time the goal was to use the sense of touch to sense an object that is in the form of a valve wheel. The scene and resulting map are shown in Fig. 6.13c-d. The wheel in the scene is 0.208m x 0.209m x 0.019m and weighs 4.536kg. This experiment was performed to see how the finger and control architecture handled round and more complex shapes. The flexibility in the joints along with the control architecture for detecting the edges allowed for the circular shape of the wheel and the inner detail to be easily reconstructed.

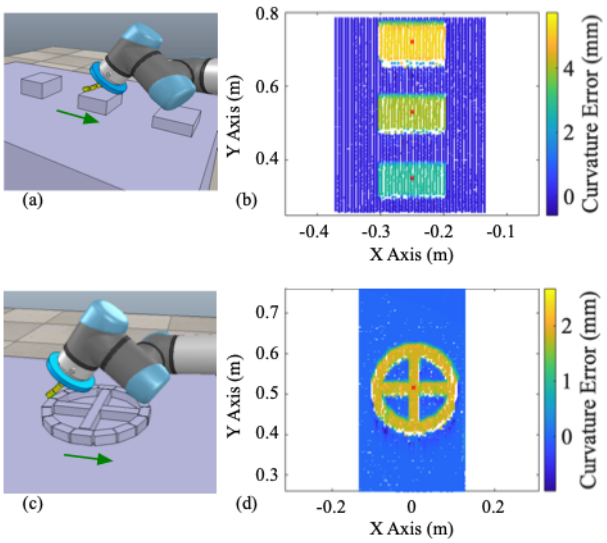

Figure 6.13: Visualization by the sense of touch: The finger is attached to a robotic arm and its first position is at the northeastern position of the table. The green arrow indicates the direction the arm moves for each pass. (a) The scene with the boxes. (b) The color coding represents the amount of tracking error, which increases in value when the finger is in contact with the boxes. The red mark shows the center of each box. (c) The scene with an object in the form of a valve wheel. (d) The color coding represents the amount of tracking error, which increases in value when the finger is in contact with the object. The red mark shows the center of the object.

Triangular curvature reference experiments: In these experiments, the tracking of the curvature reference results in the finger flexing and extending. The experiment is illustrated in Fig. 6.14$a_1 - a_4$. The arm was first positioned at the northeastern corner in front of the object

and moved in the -y-direction toward the object at a speed of $0.002m/s$. Once the object was detected (Fig. 6.14a_2), the arm stopped moving in the -y-direction and moved in the x-direction laterally along its surface at a speed of roughly $0.003m/s$ continuing to track the periodic ramp reference mimicking a tapping motion (Fig. 6.14a_2) until it no longer detected the object, see Fig. 6.14a_4. The finger was moved more slowly in this scenario in order to give the finger time to extend and not simply be dragged along the object while performing the tapping motion. In this case, the method made periodic contact with the object along the surface and indicated when contact was detected.

Two experiments were performed for a scene with one box that was $0.1m$x$0.1m$x$0.03m$ and weighed 2.5kg. The first used a periodic signal of two seconds with a threshold of $T = 3mm$. The second one used a slower periodic signal of four seconds and a threshold of $T = 2mm$. The results are shown in Fig. 6.14b_1 and Fig. 6.14b_2, respectively. In both experiments, we can see that the edge detection signal is periodic and as can be expected, the four-second period has half of the detections of the two-second signal. Since the error threshold for the slower signal was lower, it detected the box quicker. However, it also detected the box for a shorter distance since the finger was in an extension at the end of the box's surface. The threshold could be reduced slightly for the quicker period to allow for earlier detection. However, we chose to be more conservative to avoid having false detections. This would be even more valuable on a physical system where a sensor is likely to be even more noisy. Another option would be to slow down the arm motion to give the finger more time to sample the space.

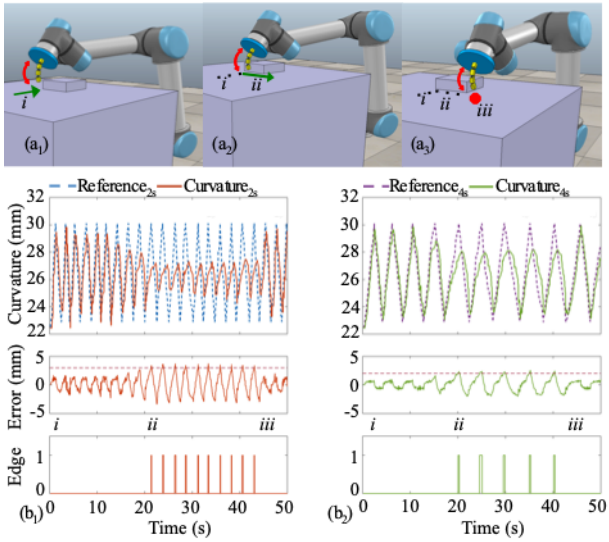

Figure 6.14: The tapping finger: the finger is attached to the robotic arm that moves toward the box on the table (*i*). ($a_1 - a_3$) Once the finger makes contact with the box, the arm moves laterally across the box in a smooth motion tapping its surface to determine where the object ends (*ii*) and it stops at (*iii*). The path of the arm is shown with the green arrow and its end is denoted with the red dot. The finger tracks the triangular reference signal resulting in flexing and extending (shown with the red arrow) while the arm is moving toward the object.

6.6 Conclusion

We presented the novel approach to enable the sense of touch for compliant soft robotic fingers. The approach is based on the reference tracking feedback control in which the tracking error is used as the measure of the finger stress resulting from the contact between the finger and an object.

For the touching tasks in which the finger had to be at the fixed position and able to drag over objects, we considered the constant value reference. For the tasks in which was necessary to tap objects, we used triangular reference signals resulting in flexing and extending the finger. By combining the two references, we were also able to achieve the firm touch between the finger and an object. The control design was based on the system identification method

in which we showed that the finger can be modeled and successfully controlled based on the second order linear model. The controller was tested in the matlab simulation and then used with the finger in the V-REP simulator. To be able to evaluate if the finger was in the contact with an object or to detect its edges based on the tracking error, we performed the experiments from which we obtained the error distribution and used them to define the detection threshold. While more complex detection may be required for more complex tasks, we also showed that the tracking error signal intensity towards the use of the sense of touch to create the map that visualizes objects in the operating space. If that is the case, object edges and object detection can be achieved using techniques from image processing.

Overall, this work shows the effectiveness of the feedback enabled sense of touch and future work can be aimed at introducing new touching tasks, their coupling with the robot arm motion, as well as developing control algorithms for the control of compliant multi-finger robotic grippers and anthropomorphic robotic hands.

Chapter 7

Proprioceptive Modeling and Control of a Soft Somatosensitive Actuator for a Kinesthetic Feedback Approach to Touch

This chapter is a preprint to the paper

- M. Boivin, K. Lin, D. Milutinović, and M. Wehner, "Proprioceptive Modeling and Control of a Soft Somatosensitive Actuator for a Kinesthetic Feedback Approach to Touch," (in prep).

7.1 Introduction

Many tasks in robotics require physical interactions between the robot and its environment. While soft robots can simplify some contact tasks such as grasping due to their compliant nature, there are still challenges in their operation related to their interactions with

objects of unknown location, shape, orientation and surface characteristics. To address these challenges, many contact-rich tasks combine multiple sensing modalities from vision and touch, and manipulators rely on processing visual data from cameras. However, the information from vision modalities could become unreliable if line of sight becomes blocked, the light source fails [68], or when dealing with transparent objects [57, 107]. In these situations, humans rely not only on visual feedback, but also on the sense of touch through proprioceptive and tactile sensory systems of their fingers. Inspired by the human sensory system, we present here the design of a dynamically based kinesthetic feedback control architecture for a finger-like soft robotic component with embedded sensors (Fig. 7.1A) to enable touch through proprioceptive sensing. By enabling the robot to have a robust sense of touch at the end-effector the robot's workspace can still be visualized when combining that data with image processing techniques without needing to rely on precise vision data.

In humans, the sense of touch is described by two modalities, *cutaneous* and *kinesthetic* [27]. The cutaneous sense of touch refers to the modality that is often thought of as haptic or tactile sensing in which the sensory input is from the receptors embedded in the skin, while the kinesthetic sense refers to the awareness of how the body is moving in space and receives sensory inputs from the receptors within muscles, tendons, and joints [27]. Proprioception on the other hand, cares only about where the body is in space, regardless of how it got into that position. When creating an analogy for robots, cutaneous sensing corresponds to direct perception of pressure or shear force due to a sensor at the point of contact (force on a finger tip due to gripping an object) [11, 12, 27, 52, 126]. Kinesthetic sense is derived from modulation of another internal parameter (increased force required at a shoulder to lift a heavy object with the hand). The latter

is the focus of this paper.

We propose here a framework for a kinesthetic touch approach based on system identification of a dynamic model for a finger-like shaped soft robotic actuator innervated with soft fluidic sensors. The proposed method for identifying the model of the finger curvature addresses environmental variations such as change in humidity and temperature as well as variations in material or human factors during the fabrication process that can have an affect on the finger dynamics. Our control architecture is a two state contact detection strategy that starts with a freely moving finger not in contact, detects when contact is made, and switches between the non-contact motion to maintaining contact. We design a reference tracking controller, which uses the finger's curvature measurement signal and enables the sense of touch. The reference tracking control allows to set the signal either to a desired value, or to follow a desired reference signal, which results in a controllable motion of the finger over configurations in the range between full flexion and extension of the finger. The tracking error of the control loop is negligible when the finger's motion is unimpeded, i.e., not in touch with any object. When the finger is in touch with an object, the error increases and its intensity measures the amount of stress on the finger. A state machine demonstrating the two state method is presented in Fig. 7.1D where the finger first starts in free displacement, i.e. not in contact with anything, and begins tracking a curvature measurement reference signal. Along with the reference signal, a second input, the touch reference, is given that determines the how much pressure to apply (light or firm) after contact is detected. The touch signal is used to determine how to change the curvature measurement reference signal. Moreover, this method of touch detection can be used for other soft robotic fingers with curvature sensors [55, 89, 128, 129].

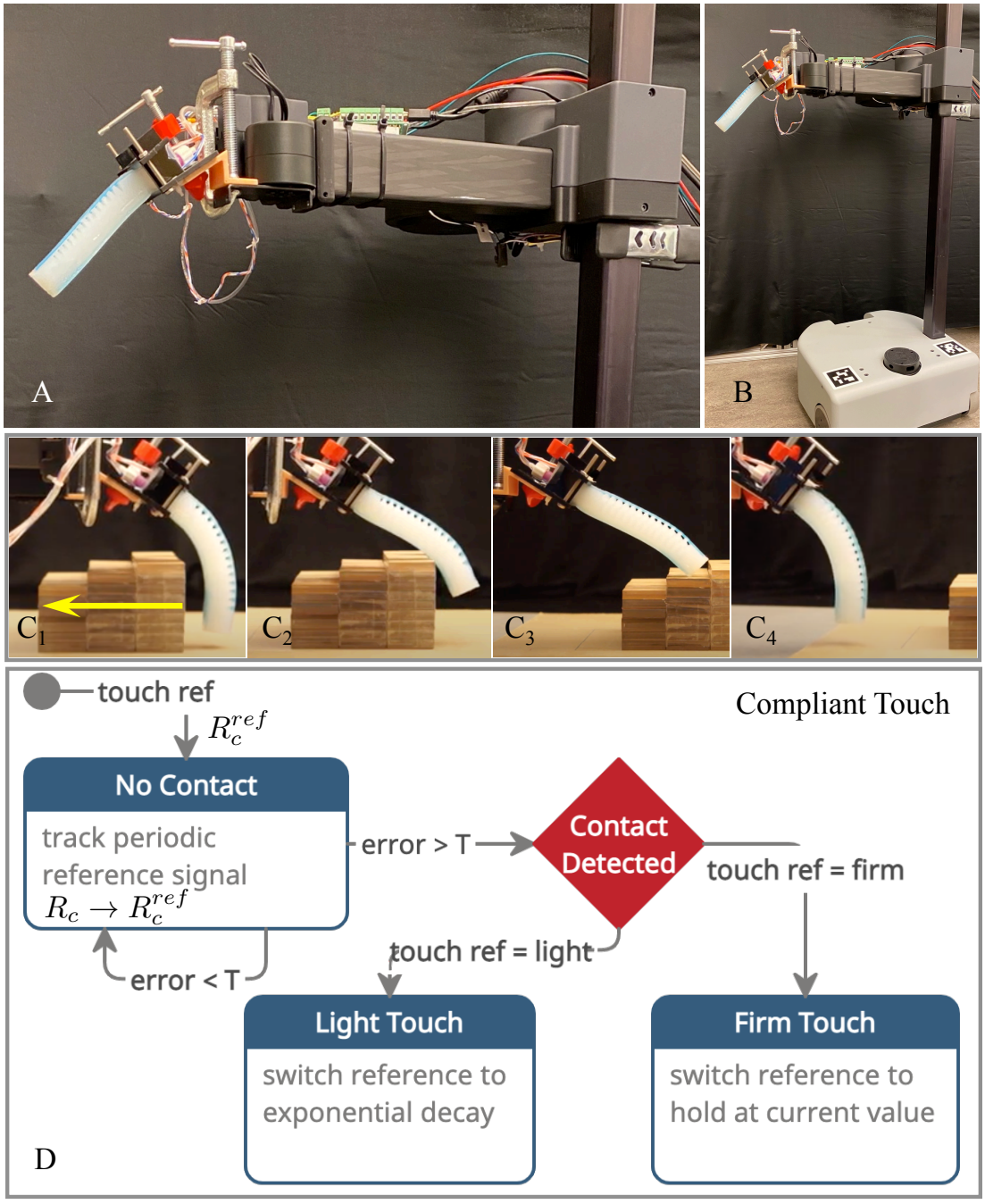

Figure 7.1: (A) The soft finger-like actuator mounted on the Stretch RE1 robot [103]. (B) The full Hello Robot Stretch RE1 robot with the finger attached. (C) An example of a contact-rich task where the natural compliance of the finger reacts as the finger is dragged across the objects. (D) The state machine for the error based compliant touch with objects.

Fig. 7.1a-b shows the soft finger attached to the Hello Robot, Stretch RE1 robot [103]. When the controlled finger makes contact with an object, if the finger is dragged (see Fig. 7.1c), the curvature is reduced as it comes into contact with an object and returns to the reference curvature once it is no longer in contact. The tracking error can be used to detect objects, their edges, and the level of the error can be used as an indication of change in object heights. Controlled dragging motion of the finger also allows for visualizing objects and their textures following similar principles that are used in atomic force microscopy [46], but at a scale of the finger's size. The presented work demonstrates our method in several examples of contact detection by one finger. However, we believe that the method presented will have even greater impact when used in a multi-finger gripper. With further research, we believe that this method will allow great advances in soft robot gripping and even dexterous manipulation.

Related work: Soft robotic manipulators have greatly improved over the last few years [67, 105, 110]. The compliant nature of soft robotic components lends itself well to manipulation and grasping tasks since the soft structures naturally form around objects where there could be uncertainty in shape or orientation and are inherently safer for fragile environments, objects [41, 72] and humans [25]. However, due to their flexibility, soft robotic components have less constrained movements and are difficult to model and control [42], especially when the soft component is in contact with objects in the operating environment [55]. Consequently, both physics-based and data-driven models have been used to control soft robotic components.

Multiple link manipulators or arms that can move in three dimensional space have complex complex nonlinear dynamics. Physics-based modeling of soft robotic manipulators has largely consisted of constant curvature kinematic models [32, 61, 124] or Lagrangian dynamics

[85]. Recently, system identification techniques are being applied for identifying models or unknown parameters [34, 85]. These methods are useful for handling any fabrication induced uncertainties. Some designs are also sensitive to environmental conditions like humidity and temperature [118] or the viscoelastic deformation [28]. Nonlinear models have been identified using methods like neural networks [20, 114] or nonlinear auto-regressive with exogenous inputs models (NLARX) [44, 58, 74, 75], and reinforcement learning [113]. The Koopman operator theory has been used as a way to construct a linear dynamical model [17] and combined with model predictive control [16]. Similar to our approach, this method gives an explicit control-oriented linear model rather than a black-box input-output mapping such as that given by neural networks.

Many control approaches have focused on position control utilizing external camera feedback [62, 81, 83, 85]. However, this limits the environment in which the manipulators can be controlled. Recent improvements in soft sensing capabilities allow for curvature and tactile sensing [114, 118, 122]. By incorporating sensors, closed-loop controllers can more easily be designed using the sensor feedback [45]. Sensors can also be utilized for designing filters for state estimation [74, 75, 77] and observers that can improve the position without the dependence on camera systems. Measurements from an embedded flex sensor were used with an extended Kalman filter to estimate the curvature of a soft finger [74, 75].

Our two state approach to detect and maintain contact between the soft finger and an object is inspired by kinematic problems [86], in which the anticipation of future relative motion between system variables is instrumental for robust decision making. The feedback control that we propose employs the control to set the compliant robotic finger curvature to a certain

position, or reference movement, which serves as a *prior* for the movement. The approach uses the discrepancy between that curvature resistance reference and the actual curvature resistance signal to infer robustly that the finger is in contact with the object, see Fig. 7.5.

In our preliminary study for the control of a soft robotic finger [13], we used a continuous-time second order controller along with a deterministic static characteristic relation between the inflation pressure and curvature. Here we use system identification to estimate a dynamic model of the finger. To achieve that, we applied an impulse to the finger from which we learn that the finger can be represented and successfully controlled by a second order discrete-time domain linear model. We use the identified model to design a reference tracking digital controller. However, in order to use the feedback controlled finger for the sense of touch, we have to evaluate the controller performance for the finger that is mounted on a robot arm. This provides us tracking error distributions for various motions of the robot arm, and thresholds that we use for edge detection, or a compliant touch by the finger.

The paper is organized as follows. Section 7.2 formulates the problem of object detection utilizing proprioceptive feedback. The finger design is presented in Section 7.3. A system identification approach for modeling the curvature of the finger is described in Section 7.4. The control architecture is described in Section 7.5, with the reference tracking edge detection defined in Section 7.5.4. Section 7.6 presents results from experiments in which the finger is used to detect object edges and visualize objects. Section 7.7 gives conclusions.

7.2 Problem Formulation

Figure 7.1 shows the finger used in our study. The finger is a pneumatic-based finger, innervated with a bio-inspired network of ionically conductive fluidic sensors [40]. The finger contains a curvature and contact sensor demonstrating proprioceptive and haptic feedback. The single input control variable for the finger is air pressure, which distributes across the inflation chambers to actuate the finger. We denote this control variable by p.

When deflated, i.e. control input $p = 0$, the finger curvature depends on its relative orientation and with respect to the gravitational acceleration. When inflated, i.e. control input $p > 0$, the finger bends freely in an arc. The degree of finger bending is measured by the embedded curvature sensor, which is designed to deform when stretched or compressed. As the finger flexes and extends the shape of the sensor changes, consequently changing the resistance of the curvature sensor (see Fig.3A).

Utilizing the curvature sensor we consider the problem of creating a robust proprioceptive signal for a soft robotic finger's sense of touch using the error of the curvature signal without a force sensor. The approach we propose is based on a kinesthetic feedback control of the finger. Figure 7.5 shows the block diagram of the feedback controlled finger, where the controller, $D_c(z)$, is a reference tracking curvature controller. Provided the controller produces the zero error tracking of the curvature reference when the finger is free to move, then an obstacle blocking the finger's motion results in a non-zero tracking error. Due to the finger's flexibility the error signal can be used as an indication of contact with an object.

To design the controller a dynamic model is desirable. However, soft robots are

inherently compliant and have many parameter uncertainties from material composition and human variables during the manufacturing process making it difficult to develop model-based kinematic and dynamic formulations.

Two problems that we consider here are (P1) the design of a feedback control for the finger curvature's sensor signal reference tracking, and (P2) the formulation of the reference curvature signal for specific applications of detecting contact with objects.

To address *P1*, we use a data-driven system identification approach for determining the finger dynamics and use the identified model for the control design. For *P2*, we demonstrate the controller's work in multiple scenarios.

7.3 Soft Somatosensitve Actuator Design

The finger (see Fig.7.2) is a pneumatic-based actuator containing embedded ionically conductive fluidic sensors. The design includes a curvature sensor for proprioceptive sensing along the dorsal side, and a force sensor for tactile perception along the ventral side, and inflation chambers for motivating the actuator, as shown in Fig. 7.2A. The actuator contains five layers constructed from both Ecoflex-0030 and Dragon Skin 10 (Fig. 7.2B) in order to increase grip strength without sacrificing sensor responsivity. The inflation chambers and force sensor are created using Dragon Skin 10, while the curvature sensor is created using Ecoflex-0030 so the actuator will maintain a curving motion when inflated. The actuator is integrated into a mount system that securely holds it to provide strain relief on the connecting wires (Fig. 7.2B). To increase the robustness and longevity of the actuator, the sensor fluid reservoirs are located under

the actuator mount. This prevents the wires, which pierce the reservoirs, from breaking due to repeated bending stress as the actuator inflated. Both sensor channels are filled with the ionic fluid 1-ethyl-3-methylimidazolium ethyl sulfate (1E3MES). A description of the electrical and air subsystems used in this work are described in Chapter 5.

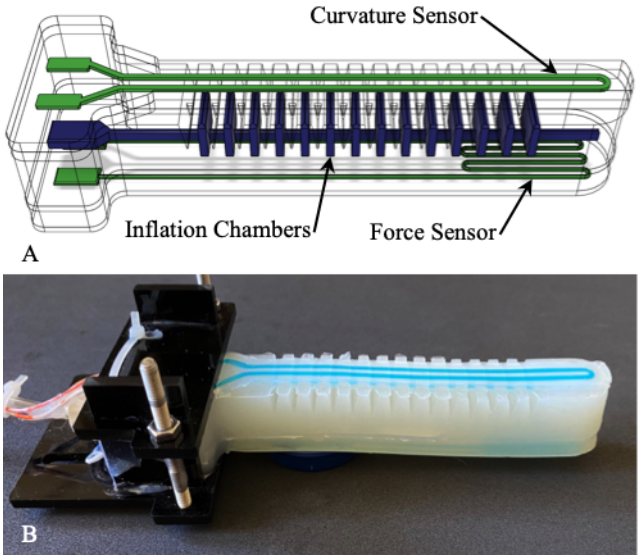

Figure 7.2: Soft robotic actuator with embedded soft sensors model. (A) The schematic of the complete finger that was developed. The curvature (top) and force (bottom) sensors are shown in green, and inflation chambers in blue. (B) The complete soft robotic actuator contained within the mount.

7.4 System Identification

To determine the approximate system model order, we performed several open loop experiments characterizing system passive dynamics. The deflated finger was tapped quickly to see the transient response to an impulse, shown in Fig. 7.3. We can see from the data that ΔR_c mimics the response of a damped harmonic oscillator. This experiment also gave some insight into the natural frequency of the finger to help determine the sampling time. The sample time

was also effected by limitations due to data collection through serial communication. Given this,

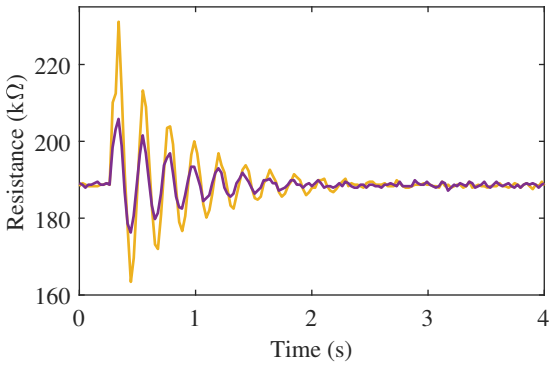

Figure 7.3: The noninflated finger was tapped to simulate an impulse response. Two tests were performed with different amounts of initial force. The first test (purple) is overlayed on the second (yellow) to show the similarities due to the finger dynamics.

for the model of the finger dynamics, we chose to identify a second order model using the linear ARX discrete time model [73]

$$y(z) = \frac{B(z)}{A(z)}p(z) + \frac{1}{A(z)}e(z), \tag{7.1}$$

where $p(z), y(z)$ and $e(z)$ are z-transform images of the discrete time control variable $p(t_k)$, the curvature measurement variable $y(t_k) = \Delta R_c(t_k)$, and the model error signal $e(t_k)$, respectively, and $k = 1, 2, \ldots$. The model sample time is $t_{k+1} - t_k = 0.15$ seconds, and $B(z)$ and $A(z)$ are polynomials of the transfer function $G(z) = \frac{B(z)}{A(z)}$ defined as

$$A(z) = 1 + a_1 z^{-1} + \cdots + a_2 z^{-n} \tag{7.2}$$

$$B(z) = b_1 z^{-1} + \cdots + b_m z^{-m}, \tag{7.3}$$

where $n \geq m \geq 1$. In the time domain, for a second order polynomial $A(z)$ $n = 2$ and using $m = n$ for the order of polynomial $B(z)$, this model results in the linear difference equation

$$y(t_k) + a_1 y(t_{k-1}) + a_2 y(t_{k-2}) = b_1 p(t_{k-1}) + b_2 p(t_{k-2}) + e(t_k), \quad (7.4)$$

which describes the relationship between the input $p(t_k)$, output $y(t_k)$ and the model error $e(t_k)$, $k = 1, 2, ...$ For this model, we estimate the vector of unknown coefficients

$$\theta = \begin{bmatrix} a_1 & a_2 & b_1 & b_2 \end{bmatrix}^T. \quad (7.5)$$

We estimate them based on the assumption that the sequence of model errors $e(t_k)$ is a zero-mean white noise.

Procedure: We identify the finger dynamics from an open-loop experiment, in which the input $p(t_k)$ is a biased pseudo random binary sequence within the pressure range defined as

$$p(t_k) = p_{bias} + p_b(t_k),\ p_b(t_k) = 19\ (kPa) \quad (7.6)$$

The bias $p_{bias} = 27.5\ (kPa)$ is added to shift the range of the pseudo random input $p(t_k)$ to be only positive since the finger does not bend backward without an external force.

$$\begin{aligned} p_b(t_k) &= \begin{cases} -p_b(t_{k-1}) & \text{with probability } \pi = 0.5 \\ p_b(t_{k-1}) & \text{with probability } \pi = 1 - \pi \end{cases} \\ p(0) &= \{8.5, 46.5\}. \end{aligned}$$

For the sequence $p(t_k)$, we collect the output data $y(t_k)$ and form the vector

$$\phi(t_k) = \begin{bmatrix} -y(t_{k-1}) & -y(t_{k-2}) & p(t_{k-1}) & p(t_{k-2}) \end{bmatrix}^T, \quad (7.7)$$

which can be used to express a one step prediction of the output $\hat{y}(t_k) = \phi(t_k)^T\theta$, for given θ. Based on this, we estimate the parameter vector θ by minimizing the sum of squares of one step prediction errors for t_k, $k = n+1, ...N$ and $N >> n$ as

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N-n+1} \sum_{k=n+1}^{N} (y(t_k) - \phi^T(t_k)\theta))^2. \tag{7.8}$$

The minimization of the prediction errors results in the solution

$$\hat{\theta} = \left[\sum_{k=n+1}^{N} \phi(t_k)\phi^T(t_k) \right]^{-1} \sum_{k=n+1}^{N} \phi(t_k)y(t_k), \tag{7.9}$$

where the right side of the solution is completely defined based on input and output data recorded at t_k, $k = 1, ...N$.

Once we have recorded the data, we compute the estimate $\hat{\theta}$ for a second order polynomial $A(z)$ $n = 2$ and using $m = n$ for the order of polynomial $B(z)$.

Given this, we obtain the second order model with parameters resulting in

$$A(z) = 1 - 0.4305z^{-1} + 0.04118z^{-2}, \tag{7.10}$$

$$B(z) = 26.85z^{-1} - 0.2633z^{-2}, \tag{7.11}$$

For an illustration, we compute the one step prediction for the second order model on new data to show that for the same input sequence, the model output closely follows the data from our experiments as shown in Fig. 7.4.

7.5 Control Architecture

The control architecture used here has been expanded upon from previous work [13]. We start with a curvature tracking controller that puts the finger in motion, shown in Fig. 7.5a.

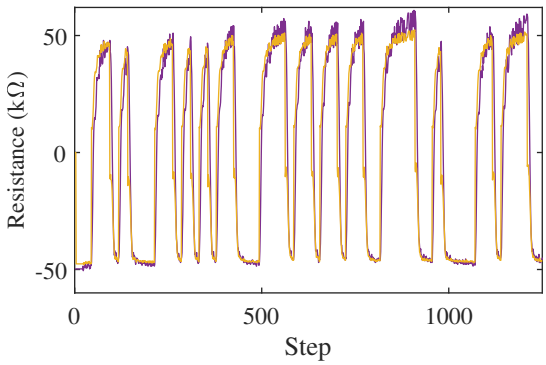

Figure 7.4: The 2^{nd} order ARX model (yellow) and measured output (purple) for the same input data.

The controller imparts that the curvature of the finger ΔR_c follows the reference ΔR_c^{ref}. If the finger is able to move freely, and makes no contact with an object in its path, we expect a small tracking error, $\Delta R_c^{ref} - \Delta R_c$. Therefore, we propose using the tracking error as a signal to detect contact of the finger with an object in the robot's workspace. Furthermore we can use this signal to not only detect contact, but also when contact is interrupted in the case where the finger is swept across the workspace and doesn't remain stationary.

7.5.1 Curvature Reference Tracking Control

We consider triangular and sinusoidal reference signals that result in the freely moving finger such that it resembles the motion of a finger sampling points in space. For tracking the triangular reference signal by the system modeled by $G(z)$ we use a unity feedback loop with the controller $D_c(z)$. From the discrete linear system control theory, we know that the steady state of the error $e_{ss} = \lim_{t\to\infty} e(t)$ for tracking the ramp reference, which corresponds to a long

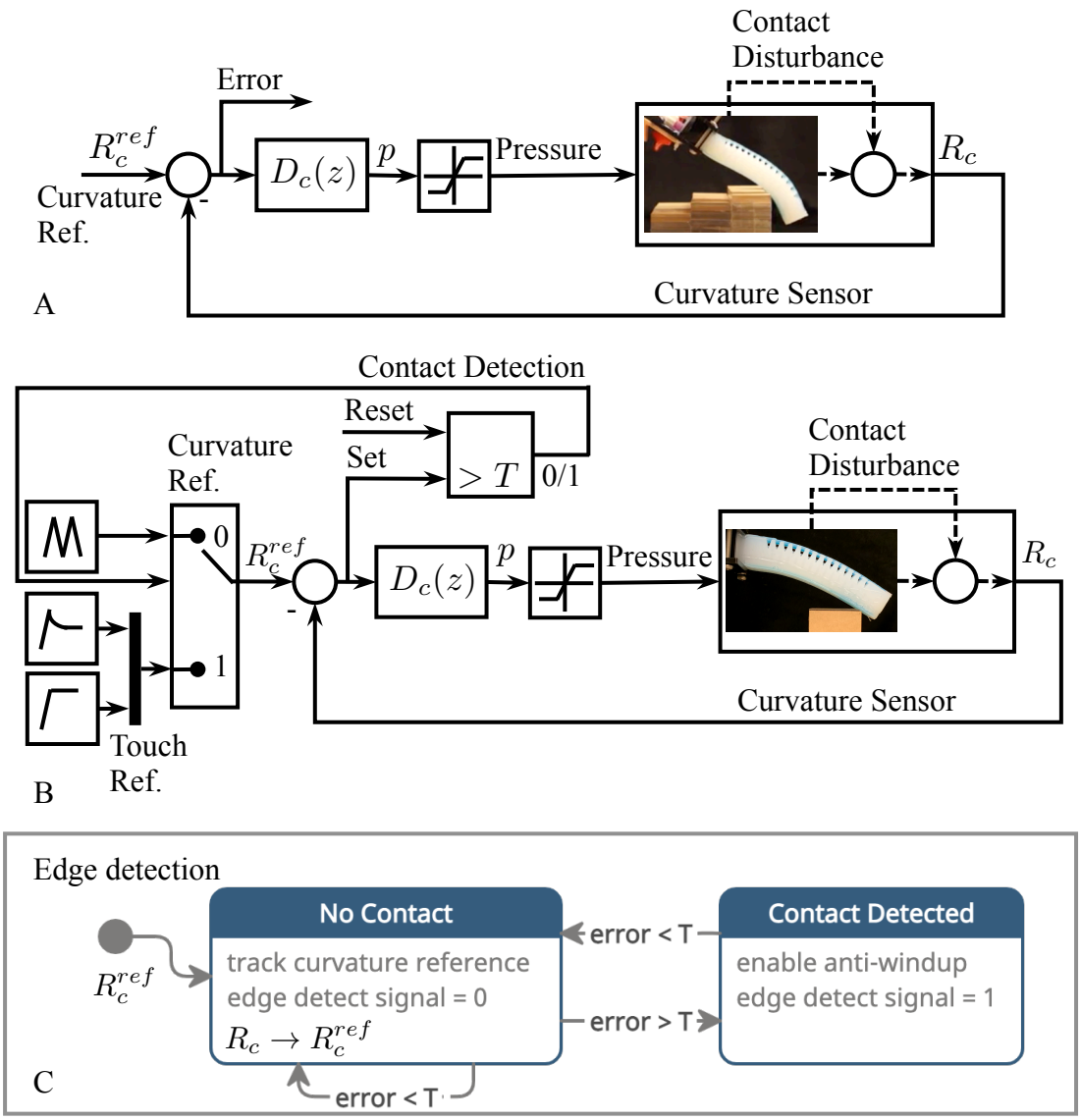

Figure 7.5: The control architecture. (A) The feedback controller D_c input is the tracking error, which is the difference between the reference and measured curvature of the finger, and the controller output is the pressure. For an unimpeded motion of the finger, the tracking error is small. If the error passes a threshold, it indicates that the finger is in contact with an object. (B) The control architecture used to maintain a compliant touch with an object. If the error passes a threshold, the reference input switches to a constant curvature to maintain a touch. Before the movement, the block that implements the detection threshold is reset to 0. (C) The state machine for the error based edge detection of objects.

increasing slope of the triangular signal is

$$e_{ss} = \lim_{z\to 1}(z-1)\frac{1}{1+D_c(z)G'(z)}\Delta R_c^{ref}(z), \tag{7.12}$$

therefore, for the zero error tracking of a unit ramp signal reference that has the z-transform $hz/(z-1)$, where h is the sample time, we need $\lim_{z\to 1}(z-1)D_c(z)G'(z) = \infty$. This can be achieved with the controller that has the form

$$D_c(z) = K_c\frac{(z-z_{c1})(z-z_{c2})}{(z-1)^2}, \tag{7.13}$$

where K_c is the controller gain and z_{c1} and z_{c2} are zeroes that are introduced by the controller. The sample time was chosen based on the natural frequency of the finger as well as limitations due to data collection through serial communication. For the loop gain $D_c(z)G(z)$, we use the discrete root locus analysis for placing the zeros z_{c1}, z_{c2} and the gain K_c. To tune these parameters, we utilized the root locus diagram in Matlab and designed for criteria of the damping ratio of $\zeta = 0.707$ while also trying to optimize rise and settling times for the step response. Our design choices were chosen to try and have a responsive controller without too much oscillations or overshoot. From these, we tune $K_c = 0.25$, $z_{c1} = 0.35$ and $z_{c2} = 0.85$. Through the root locus design process we ensure the controller meets our specified requirements for the desired system performance in tracking the ramp signal. In the time domain, this controller results in the linear difference equation

$$\begin{aligned} b &= e(k) - (z_{c1}+z_{c2})e(k-1) + z_{c1}z_{c2}e(k-2) \\ p(k) &= 2p(k-1) - p(k-2) + bhNK_c \end{aligned} \tag{7.14}$$

which describes the relationship between the residual error $e(t_k)$ and the pressure $p(t_k)$, for $k = 1, 2, ...$

7.5.2 Tracking Error Performance Experiment

The finger was attached to the end of the Stretch RE1 robot (Hello Robot, Martinez, CA) using a custom mount. The mount angles the finger 30° below the horizontal plane, as shown in Fig.7.7a. The finger was angled to increase the distance between the robot arm and objects in the scene given the range of motion of the finger. The robot has a prismatic joint that can move horizontally in and out, and a lift for moving the arm vertically up and down. Due to the finger's compliant nature, sudden motion of the robot arm's could cause dynamic deformations affecting the tracking error. The experiments were performed using both a constant curvature as well as a six second periodic signal while the arm was in motion for a 3D workspace. We tested each motion of the robot separately for a total of six experiments. In all of the experiments the finger was in the orientation where it curves down toward the ground. All motion with respect to the Stretch RE1 robot was executed at a speed of $0.01 m/s$ using a trapezoidal velocity profile to minimize dynamic deformations of the finger.

1. The curvature of the finger is maintained at a constant value while:
   (a) the prismatic arm telescopes in and out in the z-direction
   (b) the lift moves the arm up and down in the y-direction
   (c) the base drives forward in the x-direction

2. The finger flexes and extends between the curvature range for roughly a six-second period

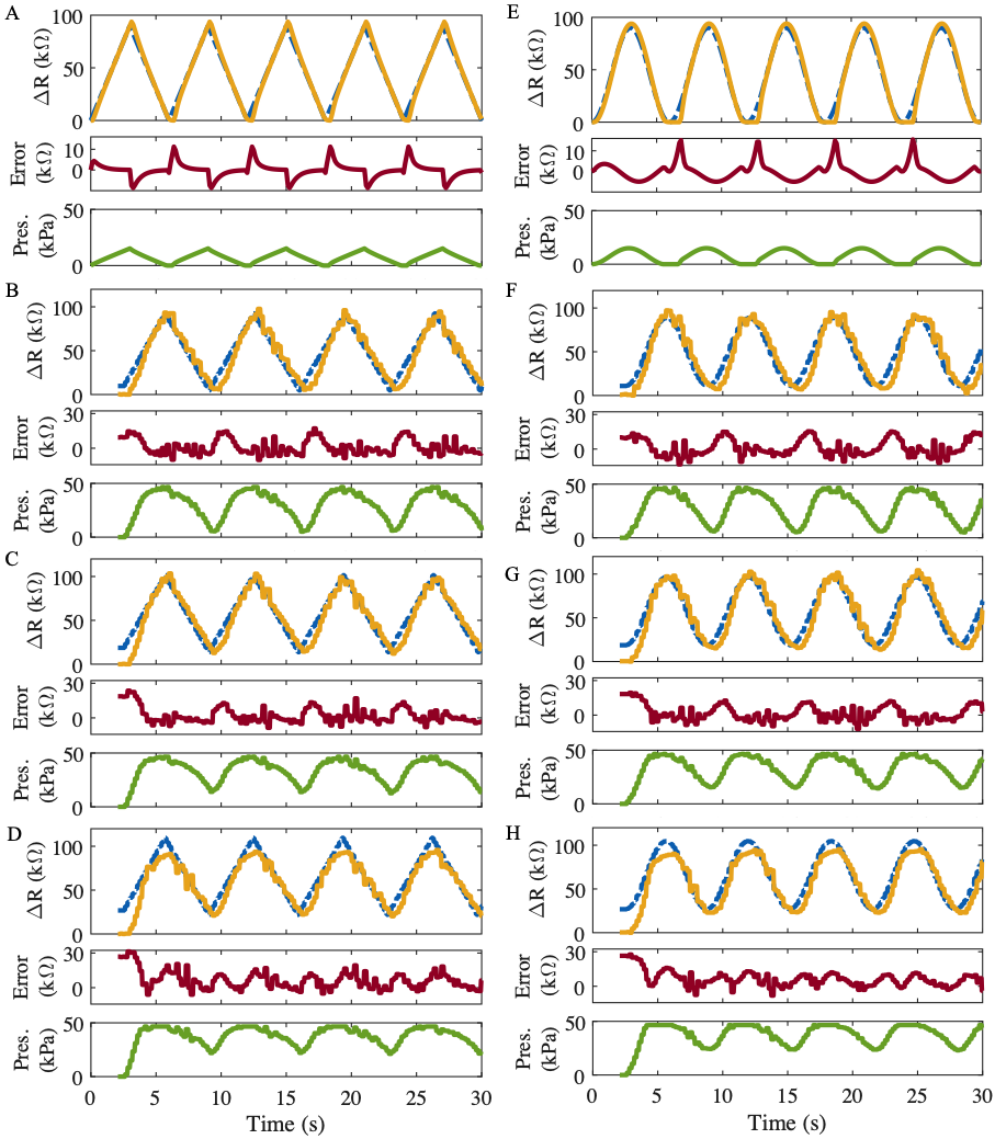

Figure 7.6: The discrete-time curvature tracking controller for the ARX model: (top panel) the reference input (blue) is the periodic signal that results in the flexing and extending of the finger; the curvature sensor measurement (yellow) tracks the reference input, (middle panel) settling to a zero error (red) steady state before changing directions; (bottom panel) the control action pressure (green) sent to the regulator. The curvature tracking controller for different orientations of the finger. Gravity in all scenarios is acting down and the pressure used to compose the reference signal is within the working range of the actuator. **(A)** and **(E)** Simulations of the discrete-time curvature tracking controller for the ARX model for a triangular and sinusoidal reference input. **(B)** and **(F)** The actuator is curling down, **(C)** and **(G)** the actuator is curling to the side, and **(D)** and **(H)** the actuator is curling up.

between extensions while:

(a) the prismatic arm telescopes in and out in the z-direction

(b) the lift moves the arm up and down in the y-direction

(c) the base drives forward in the x-direction

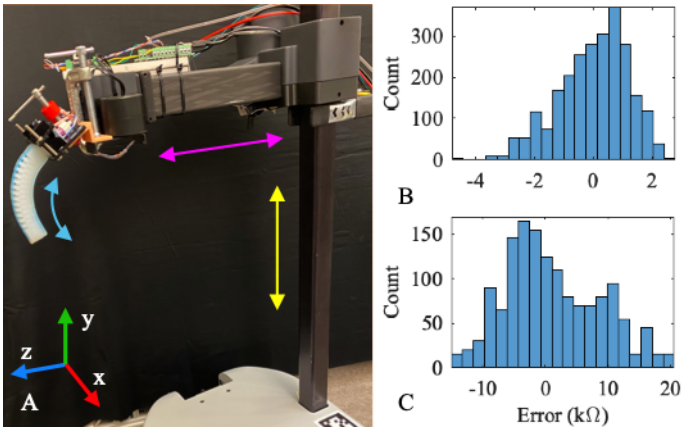

Figure 7.7: Tracking error distributions. (A) The finger in mounted to the Stretch robot. The magenta arrow shows the direction of the arm's motion along the z-axis, the yellow arrow shows the direction of the lift along the y-axis, and the blue arrow indicates the direction the finger curves. (B) Constant curvature reference: the errors were collected once the finger reached equilibrium and settled at a constant curvature reference signal. (C) The error distribution for a 6-second periodic reference signal. The histograms are obtained from errors across all the experiments that use the same type of reference signal respectively.

In the first set of experiments listed, the curvature controller starts by tracking a ramp signal in order to maintain a smooth motion of the finger until the desired curvature has been reached, at which point it holds steady. The data for the error distribution (Fig. 7.7b) only shows the values once the constant curvature reference was reached. From the first set of experiments, we find that once the finger reaches the steady state of the constant curvature reference the motion of the arm alone does not have a significant affect on the error of the system, which has an error close to zero. In the second set of experiments, the controller tracks triangular signals composed

of ramp inputs with a period of about six seconds. the error is an order of magnitude larger. The error distribution of the arm in motion for a triangular reference of six seconds is shown in Fig. 7.7c.

Through these experiments, we can see if the different types of motion of the arm affect the finger curvature and the residual error of the system. From the distributions we can conclude that when moving the robot slowly the error is largely due to the type of reference signal used, and the motion of the arm has little effect on the error. Collecting the tracking errors from the experiments, we can use the distribution of these errors (Fig. 7.7b-c) to find thresholds that indicate an impeded motion. If the error of the system is below the threshold, then it is an indication that the finger is in free displacement and has not made contact with an object in its path. The error for tracking a constant curvature (Fig. 7.7b) is an order of magnitude smaller then when tracking a periodic flexion and extension curvature reference (Fig. 7.7c). The increase in error during the periodic reference signal is due to the change in direction between flexion and extension. Since the error distributions for the two types of references vary by an order of magnitude, we chose to use two thresholds. To choose each threshold, we looked at the distributions across all experiments for each reference respectively. For a constant curvature reference, we chose a threshold $T = 4$, and for a ramp reference, we chose a threshold of $T = 22$. Each of these values are chosen to be slightly larger than the max errors of 2.7 and 20.4 seen in the error distributions as shown in Fig. 7.7b and Fig. 7.7c respectively . A look up table can be used to apply the threshold that corresponds to a desired type of the reference signal. It should be noted that while the thresholds used in our system may not be generalized to other fingers, the overall approach presented here can be applicable to the broader community using soft fingers

with curvature sensing abilities.

7.5.3 Compliant Touch Experiment

Figure 7.8 shows experiments in which the finger flexes and stops once it detects contact with an obstacle. The reference tracking controller puts the finger in motion tracking a ramp reference. This results in flexing the finger, and increasing the curvature until it has come into contact with an object. The blocked movement causes the residual error between the reference signal, R_c^{ref}, and curvature sensor, R_c, to increase. Once the threshold, T, is reached and

$$\left| R_c^{ref} - R_c \right| > T, \tag{7.15}$$

contact with the object is maintained by switching the reference to a constant curvature, which allows the finger to maintain its contact, as shown by the block diagram in Fig. 7.5b. The threshold is chosen to be $T = 22k\Omega$, which is a value slightly higher than any error value seen while the finger is in free displacement as described in the previous section (shown in Fig. 7.7C).

In these experiments, the finger base was mounted on a test setup perpendicular to the surface with the object and measured $0.057m$ above the top of the surface. The box has the dimensions of $0.03m \times 0.03m \times 0.03m$. The height between the tip of the finger and the surface was $0.0358m$, and the horizontal distance between the tip of the finger and the front edge of the box was $0.0174m$. We used the curvature controller from the previous section to actuate the finger using a triangular reference signal. Two reference switching methods were performed to show a lighter touch (just barely making contact as shown in Fig. 7.8a) versus a firmer touch (Fig. 7.8b) that might be used for grasping. In both examples the time of touch detection is determined when

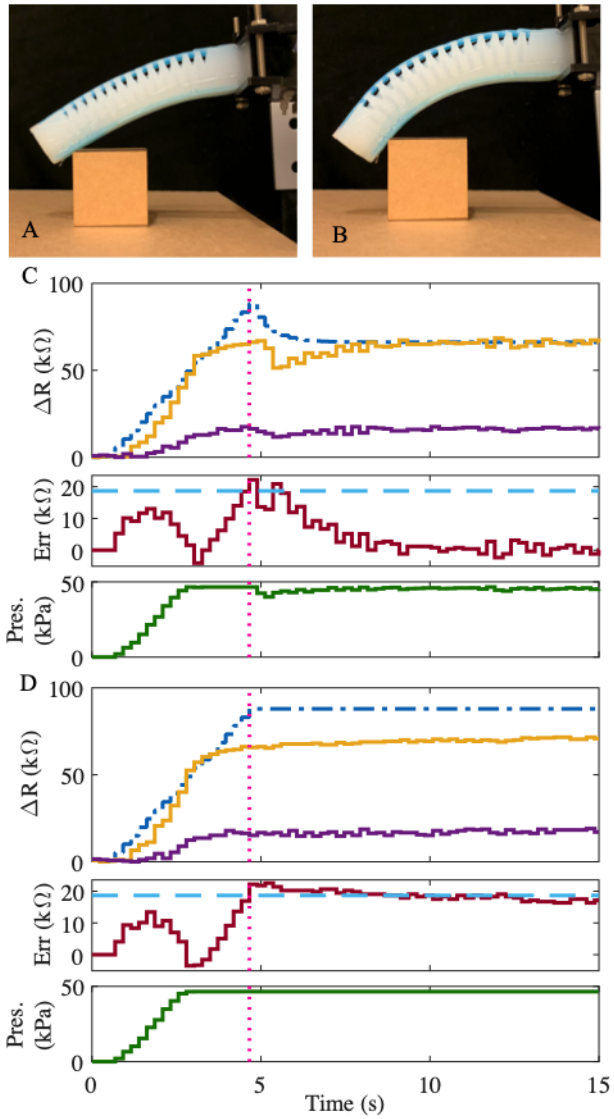

Figure 7.8: The threshold based switching for the curvature tracking controller when detecting touch. **(A)** The finger after contact is detected and the reference signal is reduced to a value to maintain a light touch. **(B)** The finger after contact is detected and the reference signal is held at a value to maintain a more firm touch. **(C)** The touch shown in (A) is detected at $t = 4.65s$ and the finger maintains a light touch on the object ($t > 4.65s$). **(D)** The touch shown in (B) is detected at $t = 4.66s$ and the finger maintains a more firm touch on the object ($t > 4.66s$). In all data plots, the reference signal (blue), the finger curvature sensor (yellow), force sensor (purple), the tracking error (red), the threshold (light blue) at $T = 22k\Omega$, the control variable, i.e. set pressure (green) are shown.

the error signal crosses the predetermined threshold defined by Eq. 7.15. This time instance is noted by the dotted pink line in the plots. In both experiments, the curvature sensor gives an indication of when the finger begins to become impeded by the block at roughly $t \approx 3.5s$ and the touch detection signal is given shortly after around $t \approx 4.6s$. Figure 7.8c shows an example of maintaining a lighter touch, which is achieved by the reference signal dropping to the value achieved by the curvature sensor when touch was detected. In order to avoid sudden change in the reference signal, an exponential decay to the new value was used to achieve a smooth transition. If the value ΔR_T corresponds to the value ΔR at the discrete time step k_T at which the threshold is reached, which is marked by the vertical dotted pink line in Fig. 7D, then the reference of the digital controller ΔR_c^{ref} is substituted with $\Delta R_c^{ref}(k) = a\Delta R_c^{ref}(k-1) + (1-a)\Delta R_T$, i.e., with

$$R_c^{ref}(k) = aR_c^{ref}(k-1) + (1-a)(\Delta R_T + R_{\mathrm{c}}(0)), \tag{7.16}$$

for all subsequent discrete time steps $k > k_T$, where $R_{\mathrm{c}}(0)$ the curvature sensor resistance for $p = 0$, $a = e^{-\lambda h}$, h is the sample time of the digital controller and λ is a constant of the exponential transition. Using a smooth transition allows for fewer oscillations as the curvature controller adjusts to tracking the new constant level ΔR_T of the reference signal.

Figure 7.8d shows the case when a firm touch is achieved by the ramp reference signal switching to maintain a constant value given by the value of the reference signal at the time of touch detection. In this case, once the threshold is reached at the discrete time step k_T, which is marked by the vertical dotted pink line in Fig. 7F, then for all time steps $k > k_T$, $\Delta R_c^{ref}(k) = \Delta R_c^{ref}(k_T)$, i. e.,

$$R_c^{ref}(k) = R_c^{ref}(k_T). \tag{7.17}$$

This creates a firmer touch since the higher reference value requires a higher pressure to maintain. While the force sensor isn't used to determine contact, it could potentially be used to determine how much force is being applied by mapping its resistance to force applied.

7.5.4 Edge Detection

Utilizing the threshold based switching rule for contact, we can extend the idea of this movement error based control even further from detecting initial touch with an object to detecting if the contact between the finger and the object was interrupted. We can utilize the same concept of crossing a threshold for detection of edges or breaking contact. An edge of a surface is detected when there is a transition in the error from the reference tracking controller crossing a threshold. While the finger is in contact with the surface, the error will be higher than the predetermined threshold since the finger is impeded. As the finger is moved across the surface, when it leaves the surface, the error will decrease back to zero since the finger is no longer blocked and able to resume tracking the original reference signal. The feedback control loop and the edge detection are depicted in Fig.7.5a and Fig. 7.5c. The edge detection block compares the tracking error to the threshold T and has set and reset inputs. Before activating the curvature controller $D_c(z)$, the output of the block is reset. Assuming the arm moves into a position in front of objects in the workspace, the front edge of an object has been detected if Eq. 7.15 holds true. Once this occurs, the output of the block is set to 1 to indicate an edge has been detected. The signal remains high until another transition is made when

$$\left|\Delta R_c^{ref} - \Delta R_c\right| < T \tag{7.18}$$

detecting that the finger has left the object either by being dragged across the surface or lifted away. The block is then reset to allow for detecting multiple edges. This combined approach is demonstrated in Section 7.6 for contact dependent tasks.

7.6 Results

We present in this section three experiments performed using the described finger actuator attached to a Stretch RE1 robot (Hello Robot, Martinez, CA). The experiments demonstrate the functionality of the combination of the proposed feedback control and control parameters from the previous section integrated with the robotic platform to perform motions for object detection with the sense of touch. In the experiments, the robot scans a small workspace in the xz-plane while the finger is held at a constant height after moving into position in front of the object(s).

Constant curvature reference experiments: In all three experiments (shown in Fig. 7.9), the arm sweeps the finger in the z-direction of the workspace at $0.01 m/s$. Once a single pass has completed in the z-direction, the base of the robot drives forward in the x-direction by half the width of the finger, $0.008m$, and the arm makes another pass along the z-direction. Due to the finger's compliant nature, its curvature naturally decreases while being dragged across the object(s). The transition from the higher to lower curvature triggers the threshold rule (7.15), indicating the finger has made contact with an object. The finger attempts to continue tracking the reference signal while the finger remains in contact with the object, and once the finger is dragged off the object, there is another transition in the curvature from low back to the reference

value triggering the threshold rule (7.18), indicating the finger has left contact with the object. In these experiments, we use the threshold, $T = 4k\Omega$, for a constant curvature determined from the tracking error performance experiments defined in Section 7.5.2.

In each experiment, three rectangular boxes are used of varying heights of 40.9*mm*, 49.3*mm* and 55.4*mm*. In the first experiment (see Figures 7.9A,D,G), the boxes are placed next to each other with no separation. In the second experiment (see Figures 7.9B,E,H), the same boxes are used, and separated 60*mm* apart. In the third experiment (see Figures 7.9C,F,I), the same boxes are separated by 90*mm*. The setup of for each experiment is shown in Fig. 7.9A-C respectively. The reference input to the control architecture is a ramp up to a constant curvature. The constant curvature value was chosen based on an inflation value of roughly 38 kPa (5.5 PSI), which inflates the finger to a curvature in the upper-middle of its working range. A single pass for each type of experiment is shown in Fig. 7.9D-F for the constant curvature.

To prevent the accumulation of error, if the maximum pressure threshold was reached (i.e. pressure was saturated), the integration and accumulation of the error was stopped using anti-windup [9]. When the finger left the object the pressure was no longer saturating, and the integration resumed. It is important to note that due to the finger's elastic nature, the transition of being dragged off an object sometimes caused oscillations that were read as multiple edges when there should have been only one. To handle this, we implemented debouncing to ensure that only a single threshold crossing would be acted upon. To debounce the signal, we checked for four consecutive samples of the edge signal to determine whether the threshold was truly crossed. Without debouncing, the oscillations may cause unpredictable results.

While we can see the shape and formation of the boxes from the images, the robot was

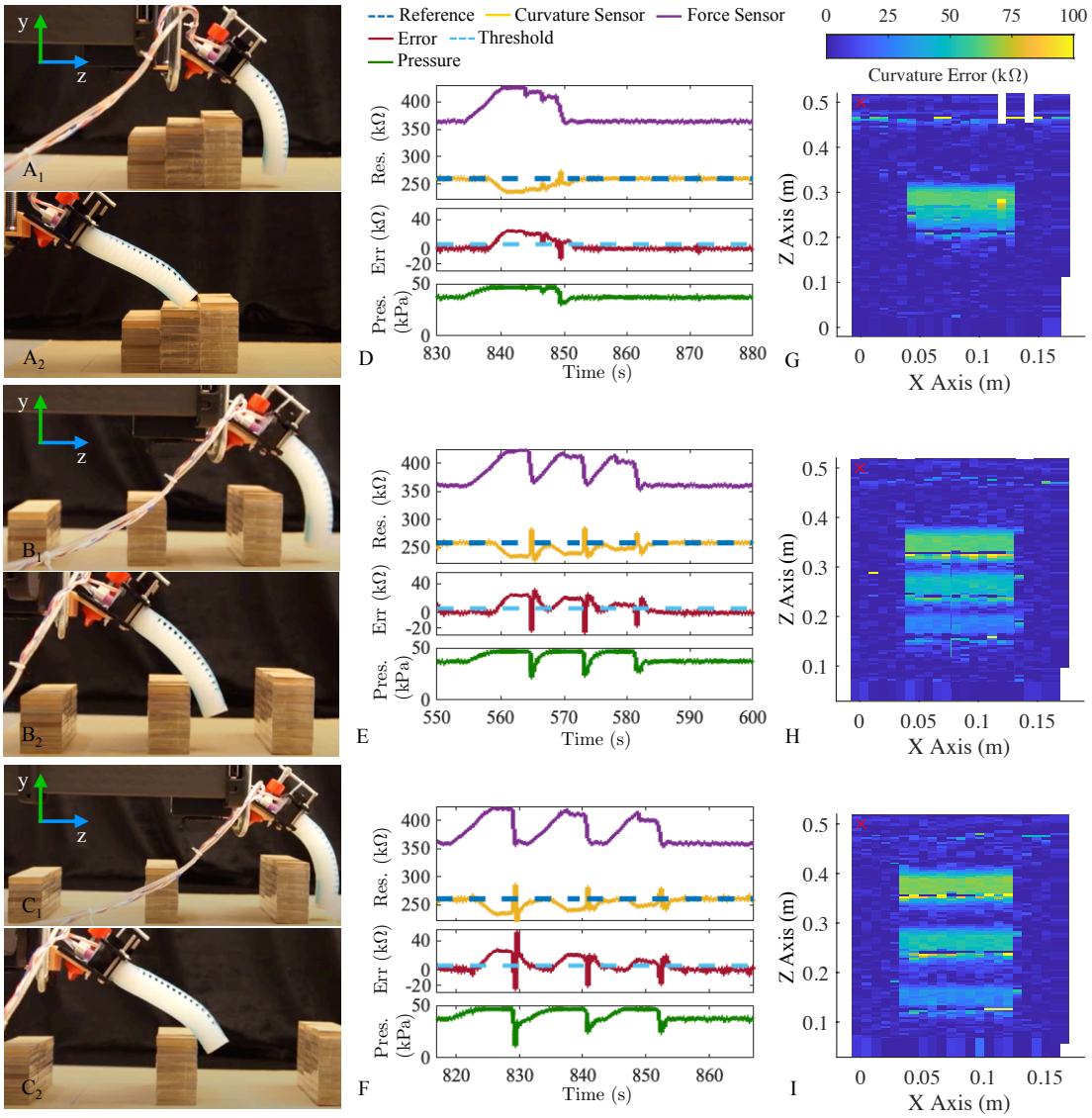

Figure 7.9: (A-C) The finger is attached to the Stretch RE1 arm that controls the dragging motion of the finger across the objects. (A,D,G) The scene with the boxes not separated. (B,E,H) The scene with the boxes separated by 60*mm*. (C,F,I) The scene with the boxes separated by 90*mm*. (D-F) The sensors, tracking error, and pressure required to actuate the finger for each scene respectively. (G-I) Visualization by the sense of touch: the color coding represents the amount of tracking error, which increases in value when the finger is in contact with the boxes. The starting position is at the upper left corner of the figure.

only given a starting location placed in front of the furthest box and given no other information about the size, shape, or orientation. From the data, we can see that the error signal contains robust information regarding the finger's sense of touch and how it interacts with the environment. The reconstruction of the scene in Figures 7.9G-H is composed only of the end-effector position and curvature error at the same time point. It not only shows roughly where in the scene the objects are, but also gives some insight into the varying heights of the boxes shown by the gradient in color. This is due to the arm maintaining the same height across all the boxes as they decrease in height. Since the finger hits at different heights on the boxes, the amount that the curvature of the finger is affected varies. This feature, along with the search pattern described, is used to create a mapping of the workspace from the tracking error signal to detect multiple objects. The result of that mapping can be seen in Figures 7.9G-I. The varying heights are indicated by the change in color which represents the amount of error seen at that point in space. The data also show an accurate reconstruction of the workspace plane, where all the boxes have a width of $0.095m$ and length of $0.03m$.

7.7 Conclusion

We presented a framework for the kinesthetic approach to enable the sense of touch for soft robotic fingers. The framework includes a method for identifying a linear dynamical model for the curvature of the finger that addresses environmental, material, and manufacturing uncertainties. The model influences the design of the reference tracking feedback controller. The sense of touch approach is based on the proprioceptive sensing capabilities of the finger

used to close the feedback loop in the controller, where the tracking error is used as the measure of the finger stress resulting from the contact between the finger and an object. We validated and demonstrated the control architecture capabilities for two contact tasks: detecting and maintaining contact. To be able to evaluate if the finger was in contact with an object or to detect its edges based on the tracking error, we performed the experiments from which we obtained the error distribution and used them to define the detection threshold.

The finger was able to determine that contact was made and maintain contact with the object by considering both a periodic flexing and extending motion and a constant curvature as possible tracking reference inputs into the curvature controller, and switching between these inputs determined by the error detection threshold. For the touching tasks in which the finger was held at the fixed height and dragged over objects, we considered only the constant value reference. While more complex detection may be required for more complex tasks, we also showed that the tracking error signal intensity can be used to create a map that visualizes objects in the operating space. Combined with image processing techniques, object edges and object detection can be achieved on the perceptual data collected from the touch exploration tasks.

Since the method uses the proprioceptive curvature sensor, it removes the requirement for the collocation between a force sensor and point of touch. In addition, the presented method does not require any specialized soft finger design requirements, and can be applied to a large number of finger designs that include a curvature sensor. The method is not dependent on external sensing technologies such as a motion capture system for the contact detection making our approach more relevant for real-world applications. We expect that the proposed touch detection method can be used with other sensing modalities either embedded in a soft finger or

external sensing for further improvements of soft robotic grippers and hands. Overall, this work shows the effectiveness of the kinesthetic feedback enabled sense of touch and future work can be aimed at introducing new contact tasks, their coupling with the robot arm motion, as well as developing control algorithms for the control of compliant multi-finger robotic grippers and anthropomorphic robotic hands for contact manipulation tasks.

Chapter 8

Concluding Remarks and Future Work

While each of the preceding papers contained conclusions on the specific area of research, here we provide some overall remarks on the dissertation and directions for future work.

This dissertation was motivated by the desire to achieve the closed-loop control of finger-like compliant actuators for touch detection and contact-rich tasks. To achieve this goal, we first had to observe how soft finger-like actuators interact with their environment. Objective (1) of the research was to develop a control architecture that utilized the embedded sensory feedback of the finger to set the finger in motion and detect contact with an object without relying on external sensing such as motion capture systems. Objective (2) was to validate the control architecture on a newly designed low-cost soft finger. Objective (3) was to improve the controller by identifying the dynamics of the finger curvature sensor. Lastly, objective (4) was to design contact-rich tasks that validated and highlighted the overall kinesthetic approach to touch framework. By not relying on external sensing our approach is more applicable to

real-world unstructured environments. We found that while soft robots have less predictable movement due to their compliant nature that the compliance overall is a benefit that coupled with embedded sensors allowed for simpler control designs to be effective methods for making contact with objects within the robot's environment. Our control designs were easily integrated into an Arduino microcontroller for both controlling the actuation pressure and collecting sensor information for visualizing the robot's workspace. However, we do recognize that by using only sensor feedback for closed-loop control means that we rely solely on sensor measurements which could be bad if the sensor becomes damaged or corrupted.

In Chapter 4 we addressed objective (1) where we proposed a control architecture that instead of using feedback control to control precisely the position of a soft finger, we used the sensory feedback to control the overall motion of the finger which allowed for touch detection when that motion became blocked by an object. Based on the sensor characteristics and under the assumption that the finger was able to move slowly we utilized a simple static characteristic relation between the sensors and the inflation pressure to inform our control design that resulted in simple deterministic linear controllers. This method allowed us to avoid using complex dynamical models that have yet to fully describe the motion of soft robots. The control architecture features a threshold based switching rule that uses the anticipation of future relative motion between the finger and an object to detect if the finger has made contact with the object. The feedback control is used to enable the finger curvature to track a reference movement, which serves as a prior for the motion. The approach uses the discrepancy between that movement and the actual one to infer robustly that the finger is in contact with the object and switches between the curvature controller to the force controller to maintain contact with the object. This work

was based on previously collected data and simulations and was yet to be validated on a physical system.

Chapter 5 addressed objective (2) and validated the general threshold based switching rule of the control architecture from the previous work. To fully validate our method, we fabricated a new somatosensitive actuator innervated with a curvature and force sensor and designed a test system. Our fabrication approach minimizes specialized equipment, bringing these devices into reach of labs with modest soft robotics infrastructure. The previous control architecture relied on both the curvature and force sensors, but this work removed the dependence on the force sensor to maintain contact with the object. Instead, the control architecture used only the curvature sensor, which decreased the importance of the co-location between the point of contact and the corresponding sensor. Our method still used the reference tracking curvature error as the signal for the threshold based switching rule, but switched between curvature reference inputs rather than between two different controllers. A ramp reference was chosen to actuate the finger for a smooth flexion motion, and once contact was detected, the reference input switched to maintain a constant curvature that allowed the finger to maintain contact with the object. Two contact motifs were explored determined by the value of the constant curvature reference, one for a light touch and one for a more firm touch. We view the firm touch as a prerequisite for the use of soft fingers for grippers with controlled grasping. We saw in the sensor characterization that the sensors seemed to be sensitive to the noise induced by the digital pressure regulator, especially at higher pressure values. We performed open-loop experiments were the actuator was given a square wave pressure input controlled by the digital pressure regulator. The pressure input would switch between a low and high pressure and hold each for a few seconds. This

experiment revealed that the digital pressure regulator is noisy and has a delay when reacting to a change in pressure. The experiment also showed that the transient response of the curvature sensor takes a longer time to reach steady state for higher pressure values then lower pressures. We can see the effects of this in the controller results. There is more of a delay on the curvature sensor response when the pressure is increasing to flex the finger compared to when pressure is being released to let the finger extend. Given the natural damping effect that the elastomer material exhibits, these results for the curvature sensor are not entirely surprising, but give some preliminary insight into the dynamics of the system, and are useful for informing future control designs.

Chapters 6 and 7 both extend the previous work and address objectives (3) and (4) but for different types of finger-like actuators, a compliant multi-link finger with flexible joints and a soft pneumatic actuator respectively. Both works employed system identification techniques for identifying a model of the finger curvature. In both cases we identified a second order linear ARX discrete time model using a biased pseudo random binary sequence as the input where the values for the input were determined by the range of possible inputs into the system. In Chapter 6 we originally identified a sixth order model, but made use of model reduction techniques, specifically pole-zero cancellation for pairs that were within close proximity to each other and where the removal of the pair still provides a good approximation of the frequency characteristics of the higher-order system. Through this model reduction process we were able to account for possibly overfitting to noise and reduce the model order. In Chapter 7 we made use of the information collected from the work presented in Chapter 5 as well as some additional informative open-loop experiments that showed the finger curvature behaves like a damped harmonic oscillator. Based

on these experiments we chose to identify the unknown parameters for a second order model. We used the corresponding dynamic models to inform our controller design which differ slightly in structure by one additional zero for the soft actuator. Each controller was tested in both free displacement and for a variety of contact tasks. The contact tasks we explored included the firm touch example from previous work, and some tasks that would allow for a visualization of the robot's workspace to be created. For these tasks we explored maintaining a constant curvature while the finger is held a constant height and dragged in a pattern that spanned the xy-plane. We found that for the compliant multi-link finger there was significant energy build up from the finger making contact with the object and then moving across the object while maintaining contact that didn't naturally dissipate when being dragged off of the object and contact ending. We accounted for this energy buildup by using an anti-windup technique where we first saved the control input required for the curvature seen when the error was very small, i.e. before making contact with the object, and then turned off integration while the finger was dragged across the object and resetting the control input to the saved value once contact was broken. In Chapter 7 where we used the soft finger, we found that this wasn't necessary. The natural damping effect that the elastomer material exhibits allowed for the controller to easily return to tracking the curvature reference after a few small oscillations upon leaving contact with an object as the finger was dragged across. Due to the finger being more heavily dampened and more compliant any built up energy naturally dissipated through the damped oscillations after the finger left the object. In both of these works, we were able to demonstrate the effectiveness of the kinesthetic feedback enabled sense of touch and that the error signal from our control architecture is quite useful in visualizing how the robot interacts with the environment and its workspace.

The ultimate goal would be to have a fully functioning soft robotic gripper that is capable of assisting humans within unpredictable, unstructured environments. The four papers included in the previous chapters represent milestones in the line of work leading to this dissertation. There are many possible directions for future work that build upon our results, and we now highlight some of them.

This work presented a method for using the error signal during touch to visualize the robot's workspace. The approach can be viewed as a step towards executing autonomous manipulation tasks. Combining different sensing modalities by fusing vision and touch could extend our approach with image processing and classification algorithms for identifying objects and being able to select from a variety of objects in the workspace would be a further step toward automated assistance in unstructured environments.

In the presented work, we only considered rigid objects for detection. It would be interesting to expand the firm touch approach to object density detection. This could likely be done by analyzing the derivative of the curvature sensor signal when coming into contact with an object. We believe that the slope would be sharper for a rigid dense object versus something that has some give to it like a plush toy. Similarly, determining if there is a relation between the size of the finger and sensitivity in the sensors for texture detection could be useful.

Data-driven system identification approaches are useful for dealing with uncertainties due to environmental variations such as change in humidity and temperature as well as variations in material or human factors during the fabrication process that can have an effect on the finger dynamics. However, combining analytical mathematical models and data-driven modeling would allow for more complex control methods to be applied. Analytical mathematical models allow

for a feedforward prediction term to be incorporated into the control algorithm. Some recent works have begun to explore this concept.

One downside to using a data-driven modeling approach is that the model is computed offline and the controller designed ahead of time based on the identified model. As mentioned previously, data-driven approaches are useful for dealing with environmental and fabrication uncertainties, but the environment could change or the elastomer material of the actuators change over time. Over time the dynamics of the system will drift from the originally identified model. Ideally an algorithm that periodically updates the model could remedy this issue. Future work could also investigate adaptive control strategies for handling the varying parameters.

Our methods show promising results for soft finger-like actuators interacting with their environment, but many assumptions were made about the structure of the environment. Some of these assumptions include not having any other external disturbances on the finger besides the object we expected to make contact with in the finger's path. To realize a truly autonomous gripper, the on board controller would need to be robust to unexpected external disturbances and react accordingly. These disturbances could represent another robot or human working in the same environment that did not give enough clearance, or an object shifting from its initial position. Given the compliant nature of soft robots, these external forces can have a huge impact on the position and motion of the robot.

While a lot of interesting work can be done using a single finger, eventually we would want to combine multiple fingers into a gripper. Future work would be in how to couple the controllers for the individual fingers to function together as a whole hand. Once the fingers are assembled into a hand and the controller coupled then different grasping strategies could be

explored as well as classifying objects based on the grasp shape.

Lastly, we believe the techniques demonstrated in this dissertation have importance outside of soft robotics. There's a possibility that our kinesthetic sensing approach for contact could be applied to rigid robotic manipulators working in contact-rich tasks. For example, this approach could be used to sense forces on an end-effector when in contact with an object by measuring the increase of torque at a joint due to the resistance without the use of force sensors. In particular, the idea of figuring out what direction a door opens by feeling the resistance on the wrist when trying to pull it in the wrong direction could be an indicator of which direction to pull the door open.

Bibliography

[1] Richard J Adams and Blake Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on robotics and Automation*, 15(3):465–474, 1999.

[2] Alin Albu-Schäffer, Christian Ott, and Gerd Hirzinger. A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *The international journal of robotics research*, 26(1):23–39, 2007.

[3] R. P. Anderson, Efstathios Bakolas, D. Milutinović, and Panagiotis Tsiotras. Optimal feedback guidance of a small aerial vehicle in a stochastic wind. *Journal of Guidance, Control, and Dynamics*, 36(4):975–985, 2013.

[4] R. P. Anderson and D. Milutinović. A stochastic approach to dubins vehicle tracking problems. *IEEE Transactions on Automatic Control*, 59(10):2801–2806, 2014.

[5] Ross Anderson and Dejan Milutinovic. Dubins vehicle tracking of a target with unpredictable trajectory. In *ASME 2011 Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control*, pages 675–682. American Society of Mechanical Engineers, 2011.

[6] Ross P Anderson, Efstathios Bakolas, Dejan Milutinović, and Panagiotis Tsiotras. Optimal feedback guidance of a small aerial vehicle in a stochastic wind. *Journal of Guidance, Control, and Dynamics*, 36(4):975–985, 2013.

[7] Ross P Anderson and Dejan Milutinović. A stochastic approach to dubins vehicle tracking problems. *IEEE Transactions on Automatic Control*, 59(10):2801–2806, 2014.

[8] Franco Angelini, Cosimo Della Santina, Manolo Garabini, Matteo Bianchi, Gian Maria Gasparri, Giorgio Grioli, Manuel Giuseppe Catalano, and Antonio Bicchi. Decentralized trajectory tracking control for soft robots interacting with the environment. *IEEE Transactions on Robotics*, 34(4):924–935, 2018.

[9] Karl J Åström and Björn Wittenmark. *Computer-controlled systems: theory and design*. Courier Corporation, 2013.

[10] Hannaford B. and Okamura A.M. Haptics. In *In: Siciliano B., Khatib O. (eds) Springer Handbook of Robotics*. Springer, Berlin, Heidelberg, 2008.

[11] Darshini Balamurugan, Andrei Nakagawa-Silva, Harrison Nguyen, Jin Huat Low, Christopher Shallal, Luke Osborn, Alcimar Barbosa Soares, Raye Chen Hua Yeow, and Nitish Thakor. Texture discrimination using a soft biomimetic finger for prosthetic applications. In *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, pages 380–385. IEEE, 2019.

[12] Chiara Bartolozzi, Lorenzo Natale, Francesco Nori, and Giorgio Metta. Robots with a sense of touch. *Nature Mater*, 15:921–925, 2016.

[13] Megan Boivin, Dejan Milutinović, and Michael Wehner. Movement error based control for a firm touch of a soft somatosensitive actuator. In *2019 American Control Conference (ACC)*, pages 7–12. IEEE, 2019.

[14] Eric Brown, Nicholas Rodenberg, John Amend, Annan Mozeika, Erik Steltz, Mitchell R Zakin, Hod Lipson, and Heinrich M Jaeger. Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences*, 107(44):18809–18814, 2010.

[15] Daniel Bruder, Xun Fu, R Brent Gillespie, C David Remy, and Ram Vasudevan. Data-driven control of soft robots using koopman operator theory. *IEEE Transactions on Robotics*, 2020.

[16] Daniel Bruder, Brent Gillespie, C David Remy, and Ram Vasudevan. Modeling and control of soft robots using the koopman operator and model predictive control. *arXiv preprint arXiv:1902.02827*, 2019.

[17] Daniel Bruder, C David Remy, and Ram Vasudevan. Nonlinear system identification of soft robot dynamics using koopman operator theory. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6244–6250. IEEE, 2019.

[18] David B Camarillo, Christopher F Milne, Christopher R Carlson, Michael R Zinn, and J Kenneth Salisbury. Mechanics modeling of tendon-driven continuum manipulators. *IEEE transactions on robotics*, 24(6):1262–1273, 2008.

[19] Raffaella Carloni, Ricardo G Sanfelice, Andrew R Teel, and Claudio Melchiorri. A hybrid control strategy for robust contact detection and force regulation. In *2007 American Control Conference*, pages 1461–1466. IEEE, 2007.

[20] Changhyun Choi, Wilko Schwarting, Joseph DelPreto, and Daniela Rus. Learning object grasping for soft robot hands. *IEEE Robotics and Automation Letters*, 3(3):2370–2377, 2018.

[21] J. Choi and D. Milutinović. Tips on stochastic optimal feedback control and bayesian spatiotemporal models: Applications to robotics. *ASME Journal of of Dynamic Systems, Measurement, and Control*, 137(3):030801–030801–10, 2014.

[22] Jongeun Choi and Dejan Milutinović. Tips on stochastic optimal feedback control and bayesian spatiotemporal models: Applications to robotics. *Journal of Dynamic Systems, Measurement, and Control*, 137(3):030801, 2015.

[23] Jean-Baptiste Chossat, Yong-Lae Park, Robert J Wood, and Vincent Duchaine. A soft strain sensor based on ionic and metal liquids. *Ieee sensors journal*, 13(9):3405–3414, 2013.

[24] Jean-Baptiste Chossat, Hee-Sup Shin, Yong-Lae Park, and Vincent Duchaine. Soft tactile skin using an embedded ionic liquid and tomographic imaging. *Journal of Mechanisms and Robotics*, 7(2):021008, 2015.

[25] Matteo Cianchetti, Cecilia Laschi, Arianna Menciassi, and Paolo Dario. Biomedical applications of soft robotics. *Nature Reviews Materials*, 3(6):143–153, 2018.

[26] Matteo Cianchetti, Tommaso Ranzani, Giada Gerboni, Iris De Falco, Cecilia Laschi, and Arianna Menciassi. Stiff-flop surgical manipulator: Mechanical design and experimental characterization of the single module. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pages 3576–3581. IEEE, 2013.

[27] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini. Tactile sensing—from humans to humanoids. *IEEE Transactions on Robotics*, 26(1):1–20, 2010.

[28] Raphael Deimel and Oliver Brock. A compliant hand based on a novel pneumatic actuator. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2047–2053, 2013.

[29] Raphael Deimel and Oliver Brock. A novel type of compliant and underactuated robotic hand for dexterous grasping. *The International Journal of Robotics Research*, 35(1-3):161–185, 2016.

[30] Cosimo Della Santina, Matteo Bianchi, Giorgio Grioli, Franco Angelini, Manuel Catalano, Manolo Garabini, and Antonio Bicchi. Controlling soft robots: balancing feedback and feedforward elements. *IEEE Robotics & Automation Magazine*, 24(3):75–83, 2017.

[31] Cosimo Della Santina, Robert K Katzschmann, Antonio Bicchi, and Daniela 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.

[32] Cosimo Della Santina, Robert K Katzschmann, Antonio Biechi, and Daniela Rus. Dynamic control of soft robots interacting with the environment. In *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 46–53. IEEE, 2018.

[33] Cosimo Della Santina, Cristina Piazza, Giorgio Grioli, Manuel G Catalano, and Antonio Bicchi. Toward dexterous manipulation with augmented adaptive synergies: The pisa/iit softhand 2. *IEEE Transactions on Robotics*, 34(5):1141–1156, 2018.

[34] Cosimo Della Santina, Ryan Landon Truby, and Daniela Rus. Data–driven disturbance observers for estimating external forces on soft robots. *IEEE Robotics and Automation Letters*, 5(4):5717–5724, 2020.

[35] Eimear B Dolan, CE Varela, K Mendez, William Whyte, RE Levey, ST Robinson, Ericka Maye, J O'dwyer, R Beatty, Arielle Rothman, et al. An actuatable soft reservoir modulates host foreign body response. *Science Robotics*, 4, aug 2019.

[36] Aaron M Dollar and Robert D Howe. A robust compliant grasper via shape deposition manufacturing. *IEEE/ASME transactions on mechatronics*, 11(2):154–161, 2006.

[37] Aaron M Dollar and Robert D Howe. The highly adaptive sdm hand: Design and performance evaluation. *The International Journal of Robotics Research*, 29(5):585–597, 2010.

[38] Khaled Elgeneidy, Niels Lohse, and Michael Jackson. Bending angle prediction and control of soft pneumatic actuators with embedded flex sensors–a data-driven approach. *Mechatronics*, 50:234–247, 2018.

[39] Khaled Elgeneidy, Gerhard Neumann, Michael Jackson, and Niels Lohse. Directly printable flexible strain sensors for bending and contact feedback of soft actuators. *Frontiers in Robotics and AI*, 5:2, 2018.

[40] Conrad Michael Esch. Construction of low cost multilayered soft robots containing embedded intrinsically soft sensors. Master's thesis, UC Santa Cruz, 2020.

[41] Kevin C Galloway, Kaitlyn P Becker, Brennan Phillips, Jordan Kirby, Stephen Licht, Dan Tchernov, Robert J Wood, and David F Gruber. Soft robotic grippers for biological sampling on deep reefs. *Soft robotics*, 3(1):23–33, 2016.

[42] Thomas George Thuruthel, Yasmin Ansari, Egidio Falotico, and Cecilia Laschi. Control strategies for soft robotic manipulators: A survey. *Soft robotics*, 5(2):149–163, 2018.

[43] Thomas George Thuruthel, Yasmin Ansari, Egidio Falotico, and Cecilia Laschi. Control strategies for soft robotic manipulators: A survey. *Soft robotics*, 5(2):149–163, 2018.

[44] Thomas George Thuruthel, Federico Renda, and Fumiya Iida. First-order dynamic modeling and control of soft robots. *Frontiers in Robotics and AI*, 7, 2020.

[45] Giada Gerboni, Alessandro Diodato, Gastone Ciuti, Matteo Cianchetti, and Arianna Menciassi. Feedback control of soft robot actuators via commercial flex bend sensors. *IEEE/ASME Transactions on Mechatronics*, 22(4):1881–1888, 2017.

[46] Franz J. Giessibl. Advances in atomic force microscopy. *Rev. Mod. Phys.*, 75:949–983, Jul 2003.

[47] Gianni Gilardi and Inna Sharf. Literature survey of contact dynamics modelling. *Mechanism and machine theory*, 37(10):1213–1239, 2002.

[48] K. I. Goldberg and R. Bajscy. Active touch and robot perception. *Cognit. Brain Theory*, 7(2):199–214, 1984.

[49] Giorgio Grioli, Manuel Catalano, Emanuele Silvestro, Simone Tono, and Antonio Bicchi. Adaptive synergies: an approach to the design of under-actuated robotic hands. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1251–1256. IEEE, 2012.

[50] Yufei Hao, Zheyuan Gong, Zhexin Xie, Shaoya Guan, Xingbang Yang, Ziyu Ren, Tianmiao Wang, and Li Wen. Universal soft pneumatic robotic gripper with variable effective length. In *2016 35th Chinese control conference (CCC)*, pages 6109–6114. IEEE, 2016.

[51] Elliot W Hawkes, Laura H Blumenschein, Joseph D Greer, and Allison M Okamura. A soft robot that navigates its environment through growth. *Science Robotics*, 2(8), 2017.

[52] Randall B Hellman, Cem Tekin, Mihaela van der Schaar, and Veronica J Santos. Functional contour-following via haptic perception and reinforcement learning. *IEEE transactions on haptics*, 11(1):61–72, 2017.

[53] Neville Hogan. Impedance control: An approach to manipulation: Part i—theory. *Journal of Dynamic Systems, Measurement, and Control*, 107(1):1–7, 1985.

[54] Cameron J Hohimer, Heng Wang, Santosh Bhusal, John Miller, Changki Mo, and Manoj Karkee. Design and field evaluation of a robotic apple harvesting system with a 3d-printed soft-robotic end-effector. *Transactions of the ASABE*, 62(2):405–414, 2019.

[55] Bianca S Homberg, Robert K Katzschmann, Mehmet R Dogar, and Daniela Rus. Haptic identification of objects using a modular soft robotic gripper. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1698–1705, 2015.

[56] Bianca S Homberg, Robert K Katzschmann, Mehmet R Dogar, and Daniela Rus. Robust proprioceptive grasping with a soft robot hand. *Autonomous Robots*, 43(3):681–696, 2019.

[57] Kaijen Hsiao, Sachin Chitta, Matei Ciocarlie, and E Gil Jones. Contact-reactive grasping of objects with partial shape information. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1228–1235. IEEE, 2010.

[58] Serhat Ibrahim, Jan Christoph Krause, Alexander Olbrich, and Annika Raatz. Modeling and reconstruction of state variables for low-level control of soft pneumatic actuators. *Frontiers in Robotics and AI*, 8:32, 2021.

[59] Filip Ilievski, Aaron D Mazzeo, Robert F Shepherd, Xin Chen, and George M Whitesides. Soft robotics for chemists. *Angewandte Chemie*, 123(8):1930–1935, 2011.

[60] S. Jadhav, V. Kannanda, M. Kang, B. Tolly, and J. P. Schulze. Soft robotic glove for kinesthetic haptic feedback in virtual reality environments. In *In: Engineering Reality of Virtual Reality (Los Angeles, CA)*, pages 19–24(6). Society for Imaging Science and Technology, 2017.

[61] Robert K Katzschmann, Cosimo Della Santina, Yasunori Toshimitsu, Antonio Bicchi, and Daniela Rus. Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 454–461. IEEE, 2019.

[62] Robert K Katzschmann, Andrew D Marchese, and Daniela Rus. Autonomous object manipulation using a soft planar grasping manipulator. *Soft robotics*, 2(4):155–164, 2015.

[63] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[64] Sangbae Kim, Cecilia Laschi, and Barry Trimmer. Soft robotics: a bioinspired evolution in robotics. *Trends in biotechnology*, 31(5):287–294, 2013.

[65] Anastasia Koivikko, Ehsan Sadeghian Raei, Veikko Sariola, Mahmoud Mosallaei, and Matti Mantysalo. Soft actuators with screen-printed curvature sensors. In *Proceedings of the IEEE SENSORS*, pages 1–3, 2017.

[66] Ying-Chih Lai, Jianan Deng, Ruiyuan Liu, Yung-Chi Hsiao, Steven L Zhang, Wenbo Peng, Hsing-Mei Wu, Xingfu Wang, and Zhong Lin Wang. Actively perceiving and responsive soft robots enabled by self-powered, highly extensible, and highly sensitive triboelectric proximity-and pressure-sensing skins. *Advanced Materials*, page 1801114, 2018.

[67] Cecilia Laschi, Barbara Mazzolai, and Matteo Cianchetti. Soft robotics: Technologies and systems pushing the boundaries of robot abilities. *Science Robotics*, 1(1):eaah3690, 2016.

[68] Michelle A Lee, Yuke Zhu, Peter Zachares, Matthew Tan, Krishnan Srinivasan, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *IEEE Transactions on Robotics*, 36(3):582–596, 2020.

[69] Ljung Lennart. System identification: theory for the user. *PTR Prentice Hall, Upper Saddle River, NJ*, pages 1–14, 1999.

[70] Steven Lessard, Pattawong Pansodtee, Ash Robbins, James M Trombadore, Sri Kurniawan, and Mircea Teodorescu. A soft exosuit for flexible upper-extremity rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(8):1604–1617, 2018.

[71] Rui Li and Hong Qiao. A survey of methods and strategies for high-precision robotic grasping and assembly tasks—some new trends. *IEEE/ASME Transactions on Mechatronics*, 24(6):2718–2732, 2019.

[72] Jiaqi Liu, Saverio Iacoponi, Cecilia Laschi, Li Wen, and Marcello Calisti. Underwater mobile manipulation: A soft arm on a benthic legged robot. *IEEE Robotics Automation Magazine*, 27(4):12–26, 2020.

[73] L. Ljung. *System Identification: Theory for the User*. Prentice Hall information and system sciences series. Prentice Hall PTR, 1999.

[74] Junn Yong Loo, Ze Yang Ding, Evan Davies, Surya Girinatha Nurzaman, and Chee Pin Tan. Curvature and force estimation for a soft finger using an ekf with unknown input optimization. *IFAC-PapersOnLine*, 53(2):8506–8512, 2020.

[75] Junn Yong Loo, Kah Chun Kong, Chee Pin Tan, and Surya Girinatha Nurzaman. Non-linear system identification and state estimation in a pneumatic based soft continuum robot. In *2019 IEEE Conference on control technology and applications (CCTA)*, pages 39–46. IEEE, 2019.

[76] Junn Yong Loo, Chee Pin Tan, and Surya Girinatha Nurzaman. H-infinity based extended kalman filter for state estimation in highly non-linear soft robotic system. In *2019 American Control Conference (ACC)*, pages 5154–5160. IEEE, 2019.

[77] Dario Lunni, Goffredo Giordano, Edoardo Sinibaldi, Matteo Cianchetti, and Barbara Mazzolai. Shape estimation based on kalman filtering: Towards fully soft proprioception. In *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 541–546. IEEE, 2018.

[78] P Maiolino, Fabia Galantini, F Mastrogiovanni, G Gallone, G Cannata, and Federico Carpi. Soft dielectrics for capacitive sensing in robot skins: Performance of different elastomer types. *Sensors and Actuators A: Physical*, 226:37–47, 2015.

[79] Carmel Majidi. Soft robotics: a perspective?current trends and prospects for the future. *Soft Robotics*, 1(1):5–11, 2014.

[80] Mariangela Manti, Taimoor Hassan, Giovanni Passetti, Nicolò D'Elia, Cecilia Laschi, and Matteo Cianchetti. A bioinspired soft robotic gripper for adaptable and effective grasping. *Soft Robotics*, 2(3):107–116, 2015.

[81] Andrew D Marchese, Robert K Katzschmann, and Daniela Rus. Whole arm planning for a soft and highly compliant 2d robotic manipulator. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[82] Andrew D Marchese, Robert K Katzschmann, and Daniela Rus. A recipe for soft fluidic elastomer robots. *Soft Robotics*, 2(1):7–25, 2015.

[83] Andrew D Marchese, Konrad Komorowski, Cagdas D Onal, and Daniela Rus. Design and control of a soft and continuously deformable 2d robotic manipulation system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[84] Andrew D Marchese and Daniela Rus. Design, kinematics, and control of a soft spatial fluidic elastomer manipulator. *The International Journal of Robotics Research*, 35(7):840–869, 2016.

[85] Andrew D Marchese, Russ Tedrake, and Daniela Rus. Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator. *The International Journal of Robotics Research*, 35(8):1000–1019, 2016.

[86] Dejan Milutinović, David W. Casbeer, and Meir Pachter. Markov inequality rule for switching among time optimal controllers in a multiple vehicle intercept problem. *Automatica*, 87:274 – 280, 2018.

[87] Dejan Milutinović, David W Casbeer, and Meir Pachter. Markov inequality rule for switching among time optimal controllers in a multiple vehicle intercept problem. *Automatica*, 87:274–280, 2018.

[88] Dejan Milutinović, David W. Casbeer, and Meir Pachter. Markov inequality rule for switching among time optimal controllers in a multiple vehicle intercept problem. *Automatica*, 87:274 – 280, 2018.

[89] John Morrow, Hee-Sup Shin, Calder Phillips-Grafflin, Sung-Hwan Jang, Jacob Torrey, Riley Larkins, Steven Dang, Yong-Lae Park, and Dmitry Berenson. Improving soft pneumatic actuator fingers through integration of soft sensors, position and force control, and rigid fingernails. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5024–5031, 2016.

[90] Alexey A. Munishkin, Araz Hashemi, David W. Casbeer, and Dejan Milutinović. Scalable markov chain approximation for a safe intercept navigation in the presence of multiple vehicles. *Autonomous Robots*, Apr 2018.

[91] Alexey A Munishkin, Araz Hashemi, David W Casbeer, and Dejan Milutinović. Scalable markov chain approximation for a safe intercept navigation in the presence of multiple vehicles. *Autonomous Robots*, pages 1–14, 2018.

[92] Alexey A Munishkin, Dejan Milutinović, and David W Casbeer. Safe navigation with the collision avoidance of a brownian motion obstacle. In *ASME 2017 Dynamic Systems and Control Conference*, pages V003T39A009–V003T39A009. American Society of Mechanical Engineers, 2017.

[93] Joseph T Muth, Daniel M Vogt, Ryan L Truby, Yiğit Mengüç, David B Kolesky, Robert J Wood, and Jennifer A Lewis. Embedded 3d printing of strain sensors within highly stretchable elastomers. *Advanced materials*, 26(36):6307–6312, 2014.

[94] Lael U Odhner, Leif P Jentoft, Mark R Claffee, Nicholas Corson, Yaroslav Tenzer, Raymond R Ma, Martin Buehler, Robert Kohout, Robert D Howe, and Aaron M Dollar. A compliant, underactuated hand for robust manipulation. *The International Journal of Robotics Research*, 33(5):736–752, 2014.

[95] Selim Ozel, Nehir A Keskin, Darien Khea, and Cagdas D Onal. A precise embedded curvature sensor module for soft-bodied robots. *Sensors and Actuators A: Physical*, 236:349–356, 2015.

[96] Yong-Lae Park, Bor-Rong Chen, and Robert J Wood. Design and fabrication of soft artificial skin using embedded microchannels and liquid conductors. *IEEE Sensors journal*, 12(8):2711–2718, 2012.

[97] Cristina Piazza, Ann M Simon, Kristi L Turner, Laura A Miller, Manuel G Catalano, Antonio Bicchi, and Levi J Hargrove. Exploring augmented grasping capabilities in a multi-synergistic soft bionic hand. *Journal of neuroengineering and rehabilitation*, 17(1):1–16, 2020.

[98] Panagiotis Polygerinos, Nikolaus Correll, Stephen A Morin, Bobak Mosadegh, Cagdas D Onal, Kirstin Petersen, Matteo Cianchetti, Michael T Tolley, and Robert F Shepherd. Soft robotics: Review of fluid-driven intrinsically soft devices; manufacturing, sensing, control, and applications in human-robot interaction. *Advanced Engineering Materials*, 19(12):1700016, 2017.

[99] Panagiotis Polygerinos, Nikolaus Correll, Stephen A Morin, Bobak Mosadegh, Cagdas D Onal, Kirstin Petersen, Matteo Cianchetti, Michael T Tolley, and Robert F Shepherd. Soft robotics: Review of fluid-driven intrinsically soft devices; manufacturing, sensing, control, and applications in human-robot interaction. *Advanced Engineering Materials*, 19(12):1700016, 2017.

[100] Panagiotis Polygerinos, Stacey Lyne, Zheng Wang, Luis Fernando Nicolini, Bobak Mosadegh, George M Whitesides, and Conor J Walsh. Towards a soft pneumatic glove for hand rehabilitation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1512–1517. IEEE, 2013.

[101] Marc H Raibert and John J Craig. Hybrid position/force control of manipulators. *ASME J. Dynamic Syst., Meas., Contr*, 1981.

[102] Steven I Rich, Robert J Wood, and Carmel Majidi. Untethered soft robotics. *Nature Electronics*, 1(2):102–112, 2018.

[103] Hello Robot. https://hello-robot.com/.

[104] E. Rohmer, S. P. N. Singh, and M. Freese. Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013. www.coppeliarobotics.com.

[105] Daniela Rus and Michael T Tolley. Design, fabrication and control of soft robots. *Nature*, 521(7553):467, 2015.

[106] Stefania Russo, Tommaso Ranzani, Hongbin Liu, Samia Nefti-Meziani, Kaspar Althoefer, and Arianna Menciassi. Soft and stretchable sensor using biocompatible electrodes and liquid for medical applications. *Soft robotics*, 2(4):146–154, 2015.

[107] Shreeyak Sajjan, Matthew Moore, Mike Pan, Ganesh Nagaraja, Johnny Lee, Andy Zeng, and Shuran Song. Clear grasp: 3d shape estimation of transparent objects for manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3634–3642. IEEE, 2020.

[108] Robert F Shepherd, Filip Ilievski, Wonjae Choi, Stephen A Morin, Adam A Stokes, Aaron D Mazzeo, Xin Chen, Michael Wang, and George M Whitesides. Multigait soft robot. *Proceedings of the national academy of sciences*, 108(51):20400–20403, 2011.

[109] Robert F Shepherd, Adam A Stokes, Rui MD Nunes, and George M Whitesides. Soft machines that are resistant to puncture and that self seal. *Advanced Materials*, 25(46):6709–6713, 2013.

[110] Jun Shintake, Vito Cacucciolo, Dario Floreano, and Herbert Shea. Soft robotic grippers. *Advanced Materials*, page 1707035, 2018.

[111] Gabor Soter, Martin Garrad, Andrew T Conn, Helmut Hauser, and Jonathan Rossiter. Skinflow: A soft robotic skin based on fluidic transmission. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 355–360. IEEE, 2019.

[112] Adam A Stokes, Robert F Shepherd, Stephen A Morin, Filip Ilievski, and George M Whitesides. A hybrid combining hard and soft robots. *Soft Robotics*, 1(1):70–74, 2014.

[113] Thomas George Thuruthel, Egidio Falotico, Federico Renda, and Cecilia Laschi. Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators. *IEEE Transactions on Robotics*, 35(1):124–134, 2018.

[114] Thomas George Thuruthel, Benjamin Shih, Cecilia Laschi, and Michael Thomas Tolley. Soft robot perception using embedded soft sensors and recurrent neural networks. *Science Robotics*, 4(26), 2019.

[115] Michael T Tolley, Robert F Shepherd, Bobak Mosadegh, Kevin C Galloway, Michael Wehner, Michael Karpelson, Robert J Wood, and George M Whitesides. A resilient, untethered soft robot. *Soft robotics*, 1(3):213–223, 2014.

[116] Ryan L Truby, Cosimo Della Santina, and Daniela Rus. Distributed proprioception of 3d configuration in soft, sensorized robots via deep learning. *IEEE Robotics and Automation Letters*, 5(2):3299–3306, 2020.

[117] Ryan L Truby, Robert K Katzschmann, Jennifer A Lewis, and Daniela Rus. Soft robotic fingers with embedded ionogel sensors and discrete actuation modes for somatosensitive manipulation. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 322–329. IEEE, 2019.

[118] Ryan L Truby, Michael Wehner, Abigail K Grosskopf, Daniel M Vogt, Sebastien GM Uzel, Robert J Wood, and Jennifer A Lewis. Soft somatosensitive actuators via embedded 3d printing. *Advanced Materials*, 30(15):1706383, 2018.

[119] Lucie Viry, Alessandro Levi, Massimo Totaro, Alessio Mondini, Virgilio Mattoli, Barbara Mazzolai, and Lucia Beccai. Flexible three-axial force sensor for soft and highly sensitive artificial touch. *Advanced materials*, 26(17):2659–2664, 2014.

[120] James Walker, Thomas Zidek, Cory Harbel, Sanghyun Yoon, F Sterling Strickland, Srinivas Kumar, and Minchul Shin. Soft robotics: A review of recent developments of pneumatic soft actuators. *Actuators*, 9(1):3, 2020.

[121] Vincent Wall, Gabriel Zöller, and Oliver Brock. A method for sensorizing soft actuators and its application to the rbo hand 2. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4965–4970, 2017.

[122] Hongbo Wang, Massimo Totaro, and Lucia Beccai. Toward perceptive soft robots: Progress and challenges. *Advanced Science*, page 1800541, 2018.

[123] Zhongkui Wang, Yuuki Torigoe, and Shinichi Hirai. A prestressed soft gripper: design, modeling, fabrication, and tests for food handling. *IEEE Robotics and Automation Letters*, 2(4):1909–1916, 2017.

[124] Robert J Webster III and Bryan A Jones. Design and kinematic modeling of constant curvature continuum robots: A review. *The International Journal of Robotics Research*, 29(13):1661–1683, 2010.

[125] Michael Wehner, Brendan Quinlivan, Patrick M Aubin, Ernesto Martinez-Villalpando, Michael Baumann, Leia Stirling, Kenneth Holt, Robert Wood, and Conor Walsh. A lightweight soft exosuit for gait assistance. In *2013 IEEE International Conference on Robotics and Automation*, pages 3362–3369. IEEE, 2013.

[126] Ruben D Ponce Wong, Randall B Hellman, and Veronica J Santos. Spatial asymmetry in tactile sensor skin deformation aids perception of edge orientation during haptic exploration. *IEEE Transactions on Haptics*, 7(2):191–202, 2013.

[127] Hongying Zhang, Michael Yu Wang, Feifei Chen, Yiqiang Wang, A Senthil Kumar, and Jerry YH Fuh. Design and development of a soft gripper with topology optimization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6239–6244, 2017.

[128] Huichan Zhao, Rukang Huang, and R. F. Shepherd. Curvature control of soft orthotics via low cost solid-state optics. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4008–4013, 2016.

[129] Huichan Zhao, Kevin O?Brien, Shuo Li, and Robert F Shepherd. Optoelectronically innervated soft prosthetic hand via stretchable optical waveguides. *Science Robotics*, 1(1):eaai7529, 2016.

Appendix A

Supplementary Materials

This section contains additional figures and the materials and methods used for designing and characterizing the actuator to allow repeatability for future researchers.

The Appendix includes:
Fig. S1. Open-loop square wave response.
Fig. S2. Rising and falling edges of an open-loop square wave response.
Fig. S3. Sensor characterization for an additional finger
Materials list
Construction Steps
Overall Framework

A.1 Supplementary Figures

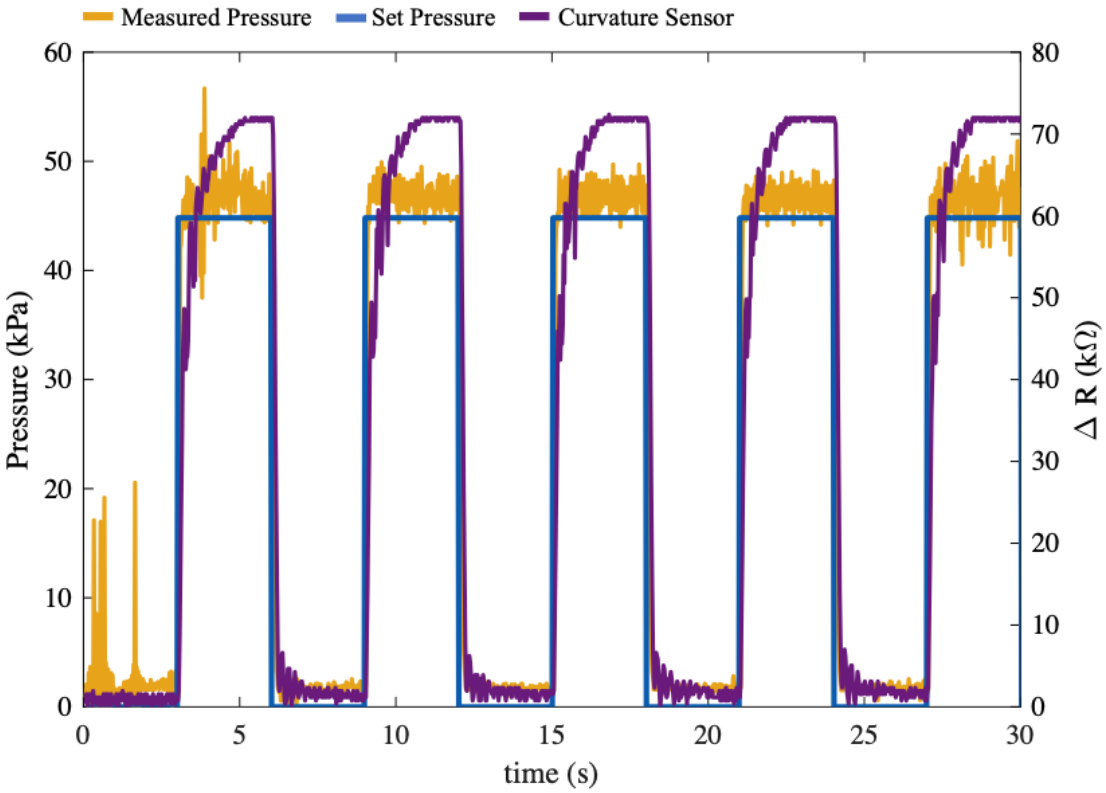

Figure S1. **Open-loop square wave response.** The noise of the digital pressure regulator can be seen in the measured pressure (yellow) and the curvature sensor (purple). The reference pressure is in blue.

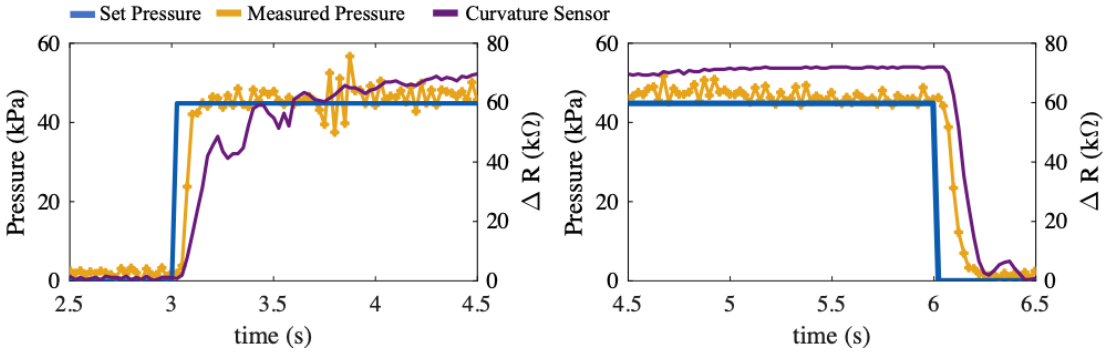

Figure S2. **Rising and falling edges of an open-loop square wave response.** The close-up of the square wave response shows the delay of both the digital pressure regulator through the measured pressure (yellow) and the curvature sensor resistance (purple). The reference pressure is in blue. **(A)** The rising edge of the square wave input shows that the curvature sensor takes about 1.5*s* to reach a steady-state value, while the measured pressure reaches the set pressure in about 0.15*s*. However, the measured pressure is noisy. **(B)** The falling edge shows the responses of both the measured pressure and curvature sensor resistance appear to be less noisy and the curvature sensor resistance reaches a steady-state value quicker.

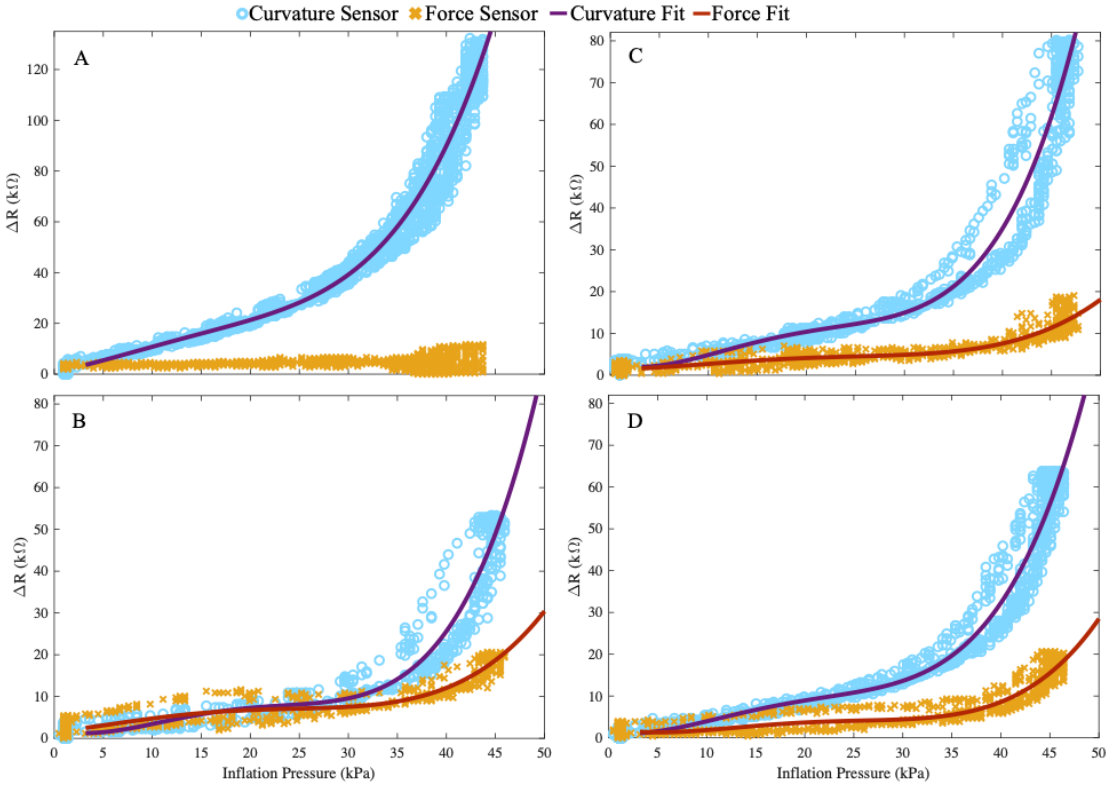

Figure S3. **Free and Blocked displacement tests for characterizing the sensors of an additional soft finger.** **(A)** The corresponding data for the finger in free displacement. **(B)** The corresponding data for blocked displacement on a flat surface. **(C)** The corresponding data for blocked displacement on a post. **(D)** The corresponding data for blocked displacement on a tube. For all data shown, the curvature sensor resistance is in blue with the fit to data shown in purple and the force sensor in gold with the fit to data shown in dark orange. Each sensor data was fit using a 4th order polynomial.

A.2 Materials and Methods

The materials and fabrication methods are further explained in more detail in [40]. Following are an itemized list of the components used within this work and the steps required to construct a similar actuator.

A.2.0.1 Materials

The total system required to control and sense the actuator consists of multiple individual pieces. The list below details the function of each individual component.

- An Arduino Nano, Atmega328p, was chosen due to its low cost, small size, and ease of programming. It can be powered by the computer it is connected to, but for accurate ADC readings, it should be provided a low drift 5V signal.
- A Digital-to-Analog Converter, MCP4725, made by Microchip and was crucial to controlling the pressure regulator which was controlled by an analog voltage. As a full range DAC with 12 bit precision, it could control the pressure regulator up to slightly over 52 kPa (7.5 PSI). It was powered by the same 5V linear regulator as the Arduino Nano.
- A Digital Pressure Sensor, Balluff BSP000W, was chosen based on it's high accuracy and the fact that it had a 0-10V output, proportional to 0-1000 kPa (0-145 PSI). The pressure sensor was powered off of a 24V DC wall adapter.
- A Digital Pressure Regulator, SMC ITV2030, provided a regulated pressure that was set by a 0-10V input. It required an air source that had a pressure greater than 103 kPa (15 PSI). It ran off of the same 24V wall adapter as the pressure sensor.
- The Relaxation Oscillator Circuit was a custom in-house PCB that was designed to measure the impedance of the sensor fluid in the actuator without hydrolyzing the fluid by using small voltage AC current. It measured the impedance of both the top and bottom sensors and converted that to a digital frequency that was easily measured by the Arduino Nano.

Multiple materials were used to create the SSAs. The materials are listed below while the specific quantities and mixing procedures are described in the construction steps section.

- The curvature sensor was made from parts A and B of Ecoflex 00–30 in a 1:1 ratio.
- The actuator air chambers and force sensor were made from parts A and B of Dragon Skin 10 in a 1:1 ratio mixed with Smooth On's Slo Jo and Silicone Thinner.
- The sensor channels were filled with the ionic fluid 1-Ethyl-3-methylimidazolium ethyl sulfate (1E3MES)
- An enclosed 3D printed SLA mold was used to make a more even, flat surface along the top chambers of the actuator where the top curvature sensor was attached. An open mold allowed a meniscus effect to form along certain sections of the top of the mold. This led to an uneven surface which made attaching the curvature sensor more difficult.

A.2.0.2 Construction Steps

The build procedure was designed through an iterative process taking inspiration from the literature. It eventually evolved into a multi-step process listed below.

1. The actuator chamber and force sensor are created using Dragon Skin 10. The elastomer was created by mixing together 20 grams of Dragon Skin 10 Part A with 20 Grams of Part B, and adding in 1 gram of Slo Jo and 4 grams of Silicone Thinner (both from Smooth-On Inc., USA). The Slo Jo helped to increase the pot life of the uncured silicone and allowed enough time to degas the silicone. The silicone thinner made degassing substantially easier.

2. We made sure the uncured silicone was effectively degassed either using a vacuum chamber or a degassing mixer system. Once the silicone was degassed, the Dragon Skin 10 mixture was poured into both the actuator chamber mold and the the bottom force sensor mold. Our actuator chamber mold required being clamped together. The bottom sensor mold can be seen in Fig. 2D. The molds were placed in a vacuum chamber to degas. Leftover Dragon Skin 10 mixture should be reserved.

3. While the Dragon Skin 10 parts were degassing, which can take up to 20 minutes, the Ecoflex 00-30 parts were made. Only the top sensor was made out of Ecoflex 00-30. The silicone was mixed using 10 grams of Ecoflex 00-30 Part A and 10 grams of Part B. Ecoflex does not need any mold thinner because of its lower viscosity. The Ecoflex mixture was poured into the top sensor mold, shown in Fig. 2E, and placed in a vacuum chamber to degas.

4. Once bubbles had stopped coming out of any of the molds, they can be removed from the vacuum chamber. We then topped off any lost uncured silicone so that it was level with the top of the mold. For the top and bottom sensor molds, a piece of acrylic that is slightly bigger than the mold itself was gently placed on top of the mold going from back to front. This served as a lid for the molds and helped to create a very smooth top surface for the top and bottom sensor. They were placed in a 60°C oven for about 4 hours to fully cure.

5. Silicone rubber skins were used instead of molded silicone pieces to seal in the sensor and actuator chamber parts. While the sensor and actuator chamber parts were curing, the silicone skins were made. Two total skins were needed for one actuator, one of Ecoflex 00-30 and one of Dragon Skin 10. The silicone for the first skin was made by combining 15 grams each of Dragon Skin 10 A and B, and add 2 grams of mold thinner. After mixing it needed to be degassed. The other skin was made by mixing 15 grams each of Ecoflex 00-30 A and B. The Ecoflex mixture didn't need any thinner as the viscosity of unaltered Ecoflex is similar to that of the Dragon Skin 10 with thinner. Once both are mixed and degassed, they can be spun coat onto acrylic plates. Either the Dragon skin or Ecoflex mixture was deposited onto the middle of the acrylic plate. Then the plate was spun at 100 rpm for 70 seconds. This created a smooth, even layer. This was repeated with the remaining uncured silicone, which should result in 2 plates of uncured silicone, one composed of Dragon Skin and one composed of Ecoflex. These were then cured in a 60°C oven, and were ready to be used in an hour.

6. Once everything had cured, the final assembly of the individual components was constructed. The same procedures as above were followed but only a total of 15 grams each of the Dragon Skin mixture and the Ecoflex mixture was needed to create a layer of glue. Once these were mixed and degassed, the same silicone mixture was poured onto the middle of the acrylic disk that was covered with the silicone skin, making sure to match the uncured silicone with its same cured silicone counter part. These disks were spun at 1800 rpm for 60 seconds. This created a very thin uncured silicone glue on top of the cured silicone skin. This process should be done for both of the plates.

7. The molded parts could now be glued to their corresponding skin. To do this, a molded part was gently placed down on the uncured silicone glue, starting at one end of the part and moving toward the other. Both the actuator chamber piece and the bottom sensor were laid onto the Dragon Skin 10 skin and the top sensor piece was put on the Ecoflex skin. These were left to cure at room

temperature. Curing them in an oven could lead to microscopic air bubbles, caught between the part and the skin, expanding due to heat and affecting the integrity of the seal.

8. After 2 hours the parts were sufficiently glued to their respective skins. The complete parts were cut out of the main skin using an x-acto knife. A visual check of all of the sensor and air channels was performed. If any of the channels looked filled in, that part was discarded. If any parts don't appear properly sealed, then a needle containing uncured Ecoflex 00-30 was inserted into the part where the two pieces were not glued together effectively. Silicone was injected to spot fix small problems.
9. To attach the bottom force sensor to the actuator chambers, 20 grams of mixed and degassed Dragon Skin 10 was spread onto the top of the bottom force sensor using a wooden applicator. The actuator part was then gently placed on top and was prevented from sliding out of alignment with the sensor using wooden sticks that were pressed into the surface that the parts were stacked on. Everything was then placed into the vacuum chamber to remove air bubbles between the bottom force sensor and the actuator. After the air bubbles had stopped coming out the joined actuator and force sensor was put in the oven to cure for two hours.
10. Next, the top curvature sensor needed to be attached to the combined actuator and force sensor. The previous step could not be used since uncured resin would could drip down and cure on top of the actuator chambers causing an uneven expansion of the chambers. To prevent this, an acrylic plate with a cutout that exactly matches the top curvature sensor was attached to an acrylic disk using double sided tape. The top curvature sensor was then gently placed inside of the cutout, and the whole apparatus was placed on the spin coater. A bead of uncured Ecoflex 00-30 was put down the length of the top sensor, and spun coat for 1 minute at 70 rpm. This applied an even layer of uncured Ecoflex on the whole curvature sensor surface. With the top sensor still inside of the acrylic cutout, the assembled actuator and bottom sensor was placed topside down onto the curvature sensor. This was then placed in an oven to cure for 2 hours. Doing this completed the actual construction of the whole actuator but does not include adding the sensor fluid and mounting the actuator.
11. To flood the sensor channels with the ionic fluid 1-Ethyl-3-methylimidazolium ethyl sulfate (1E3MES), two syringes were prepared. One contained 1E3MES and the other contained uncured Ecoflex 00-30. We added food dye to the sensor fluid to make the sensor ink, which was normally translucent, more visible. This had the added benefit of making leaks much easier to detect during development. The first step to flood a sensor was to place a hollow needle into one of the sensor reservoirs, which allowed the air to escape when the sensor fluid was injected. Using the needle filled with 1E3MES, the fluid was carefully injected into the sensor reservoir opposite the one containing the hollow needle. The fluid would slowly fill the entire length of the sensor channel. The sensor fluid continued to be inserted until it came out of the hollow needle. All air bubbles needed to be removed with an empty needle before removing both the 1E3MES syringe and the hollow needle. The sensor channel was now filled with sensor fluid, and had to be sealed properly otherwise it would leak out over time.

    To seal off the sensor channels, a silicone cap was placed inside of the sensor fluid reservoirs. After completely filling the sensor channels with fluid and removing all air bubbles, the syringe filled with uncured Ecoflex 00-30 was inject into the end of the sensor fluid reservoir where the needle that had filled the sensor pierced through the silicone. Enough Ecoflex 00-30 was injected such that about half of the reservoir was filled with silicone and half was filled with sensor fluid. This was then cured at room temperature for 2 hours.
12. After the sensor channels are filled, the actuator was mounted and electrically wired. The mount was made using laser cut acrylic pieces. It featured holes to allow a bolt to clamp the actuator,

slots to allow cable ties to provide strain relief to the sensor wires, and finally a set of slots in the back of the bottom part of the mount that would allow it to be bolted onto a structure. The mount was installed by inserting two long M3 bolts through the two holes on the tabs on both the top and bottom pieces. This served to effectively clamp the actuator in the mount. Next, two cable ties were looped through the holes directly behind the actuator, these were used for strain relief. Four, 24 gauge, solid core wires, about each 3 feet long were wrapped around the cable ties before poking one into each of the separate sensor fluid reservoirs. A large gauge needle was inserted into the back of the actuator's air channel.

13. If not resealed, the new holes created by the wire will allow the sensor fluid to leak out of the actuator. To circumvent this, the mount was designed to form a box around the end of the actuator, and allowed uncured elastomer to pool and cure on the back end of the completed actuator. The final step was to place the actuator vertically in an oven with the mount on as well, and pour uncured Ecoflex 00-30 into the end of the mount. Some of the silicone was reserved to pour in incrementally due to the elastomer leaking out of the mount.

A.2.0.3 Framework Components

The kinesthetic framework is presented bellow as a step by step process.

1. After an actuator is decided upon, the curvature sensor needs to be characterized for both free and blocked displacement to determine that its relation to increase in bending is monotonic and there is a detectable difference between free and blocked displacement for use in the error threshold switching rule.
2. Next we identified a data-driven model for the curvature sensor. In Chapters 4-5 a polynomial fit was found for the static characteristic relation and used to inform our controller, which is explain in more detail for each chapter. In Chapter 6 a system identification approach for an ARX model is presented that shows how to check for model reduction if the order of the model is unknown beforehand. In Chapter 7 we determined through an impulse response that a second order model would be a good approximation for our ARX model.
3. After a model was determined, we used root locus analysis in the MATLAB control system designer to find a linear controller and tuned the parameters with criteria of the damping ratio of $\zeta = 0.707$ on the root locus diagram while also trying to optimize rise and settling times in the step response. The order of our controller was chosen based on the type of reference inputs along with the curvature model. Since we wanted to track ramp references with as close to zero tracking error as possible we needed a controller with a double integrator.
4. We first verified the controller in Simulink using the controller and model simulating a periodic signal for free displacement curvature reference tracking. We used both triangular and sinusoidal signals for comparison. After verifying the controller in Simulink, the controller was converted into a difference equation to be used in a C program with an Arduino Nano.
5. For the real soft finger, the controller was next validated on the physical system using the setup described in Chapters 4 and 7. Results from the reference curvature tracking in free displacement for different orientations of the finger was collected to analyze and compare the effects of gravity on the soft finger and sensor. There may need to be iterations on steps 3-5 if performance is not satisfactory.
6. Next tracking error performance experiments were performed to find the threshold for the error based switching rule for our contact detection method. Experiments may depend on the robot

system the finger is mounted on. In Chapter 6 the finger was mounted on an arm with revolute joints, while in Chapter 7 the finger was mounted on an arm with prismatic joints. In both cases experiments were performed that required the arm to move in a 3D workspace to collect the error tracking data. We also performed experiments for the different type of reference signals (constant curvature and periodic flexion-extension motion) we were interested in using for our system. The data collected for these experiments are used to form the error distributions, which inform the threshold for contact detection. A threshold should be chosen such that it is higher than any error seen during the experiments.

7. The error threshold(s) chosen are used in the overall control architecture for contact dependent tasks to determine when contact has been detected. Anti-windup should also be included as part of the overall control architecture to avoid integrating the error once contact has been detected. We have given examples of some contact tasks in Chapters 4-7.

8. For contact tasks that involve scanning the workspace. The data for the end-effector position along with the error values can be combined to visualize the workspace.