Improving Wearable Devices for Virtual Reality using Soft Robotics

Pornthep Preechayasomboon

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

Doctor of Philosophy

University of Washington

2022

Reading Committee:
Eric Rombokas, Chair
Krithika Manohar
Jeffrey I. Lipton

Program Authorized to Offer Degree:
Mechanical Engineering

University of Washington

Abstract

Improving Wearable Devices for Virtual Reality using Soft Robotics

Pornthep Preechayasomboon

Chair of the Supervisory Committee:
Eric Rombokas
Mechanical Engineering

With the ever growing popularity of wearable devices, virtual reality and augmented reality, human-computer integration is not far from our reach. Wearable devices are meant to be worn as an extension of the human body. Once worn, the wearer's skin, tissue and sensory organs are coupled with the device, allowing the device to sense and act upon the body. These are similar in concept to how a soft robotic system may sense and react upon its environment, informing us that wearable devices share much of the same design space and challenges with soft robotics.

This dissertation therefore presents the building blocks for future wearable device designs that are informed by soft robotics. We first introduce soft robotics in general and a fabrication method that demonstrates how soft and rigid components can combine and benefit from one another to create new materials and more complex designs. We also look into how a soft robotic system achieves proprioception, or how a robot knows its current state, without the need for external sensors. We then introduce wearable devices that work in concert with the sensory organs underneath our skin and our vision to create haptic illusions of contact and force. Yet, an underlying problem for both wearable devices that read biological signals and provide haptic feedback is that they are useless if contact is lost with the body. Therefore, lastly, we present a soft robotic system for wearable devices that can sense this loss of contact and correct for it.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to thank my PhD advisor, mentor and fellow virtual reality geek, Dr. Eric Rombokas. Your willingness to allow me to explore whatever piques my interest has been instrumental to building up the skills and work I have today. Your creativity knows no bounds and have always enlightened me every moment I've been stuck. I would like to thank the members of my supervisory committee, Dr. Jeffrey Lipton, Dr. Krithika Manohar and Dr. Shwetak Patel, for their support and meaningful insights and discussions. I would like to thank Wanwisa for always being there for lost mechanical engineering students like me —your guidance has been monumental. I would like to thank members of the Rombolabs, as well as folks at the Reality Labs, both past and present, for the immense help, encouragement and mentoring they've provided over the years. And lastly, I would like to thank my family and friends for being my anchors and mentors throughout my graduate life.

DEDICATION

This dissertation is dedicated to my parents,
Dr.Apiruck Preechayasomboon and Ms.Nittaya Pasamontri.

Chapter 1

INTRODUCTION

Wearable devices often require flexibility and conformability while maintaining functionality —concepts that are built-in to soft robotic design, fabrication, and control. Once a wearable device is worn on the body, the extent of the device reaches beyond the embodiment of the device itself. The wearer's skin, hair, underlying tissue, nervous system, and sensory organs are now part of the device's system that can be sensed and acted upon, much like a soft robot and its environment. These concepts inform us that wearable devices should employ soft robotic technologies due to the similarities in challenges and the shared design space.

Wearable devices for human-computer interaction (HCI), can be used for either input, such as biomonitoring devices and sensorized insoles, output such as earbuds and glasses with displays, or both, such as smartwatches and augmented reality devices [124]. With skin covering most of the human body, these interfaces often partially or wholly mechanically couple with skin. Much like how clothing and accessories must be tailored for each person, wearable devices must have the same flexibility in order to achieve not only comfort but the intended functionality. This flexibility presents engineering challenges where devices must be soft and conformable yet provide electrical and mechanical stability for inputs, such as sensors, and outputs, such as haptic actuators. For instance, in a smartwatch body that notifies the wearer by vibrotactile stimulation, simply losing contact with the skin is enough to significantly deafen the stimulation. In a wearable system for surface electromyography (EMG), used for sensing the wearer's muscle activity though the skin, the electrodes create an electrical circuit with the skin and underlying tissue. Muscle activity is derived from the signals that are generated by chemical processes in muscle fibers that propagate through the

tissue up to the skin's outermost layer and the electrode [86]. Thus, any changes in the body can affect the overall electrical path and the system as a whole.

The field of soft robotics, in general, aims to create robots that are "soft" in their mechanical structure, meaning that they are elastic, conformable, and designed to handle high deformations. These properties typically give rise to robots that must embrace their inherent nonlinear response to changes from internal and external factors. Much like in biological systems, e.g. proprioception, overcoming this limitation requires a high degree of integration between sensors and actuators. In some cases, these two components are one and the same [45, 110]. Moreover, both components are often highly integrated with the fabrication process of soft robots [44]. Most importantly, all these factors build up to the challenges in controlling soft robots, as traditional controllers designed for rigid robotics with clear sensing and stiff actuators seldom work with their soft robotic counterparts, which leads to a vibrant research area where novel methods for soft robotics are constantly being proposed [25].

Looking beyond human-computer interaction, we are at the advent of a paradigm shift towards human-computer integration. Human-compatible technology [91] has been brought up as a major challenge moving forward with conformability, biocompatibility and personalization being key points to solve for future wearable devices —issues that are concurrently investigated in soft robotics research. The inherent soft, flexible sensors and actuators of soft robots make them ideal for integrating into wearable designs. Many have focused on providing assistance for joints in the hands, arms and legs in the form of soft exoskeletons [135], also known as soft wearable robotics. Some target the application of haptics due to their ability to closely match mechanical impedance and inherent comfort [155, 163, 78]. Some use soft robotic-like technologies for developing wearable sensors used for biopotential monitoring [1], usually in the form of textiles, or used for gross movement measurements [85]. These systems, if well integrated with each other, provide a strong foundation for driving the development of wearable devices in the near future.

This dissertation is organized into the following: **Chapter 2** serves as an introduction to soft robotics with a focus on fabrication and introduces a novel method, *Negshell Casting*,

Figure 1.1: An overview of the chapters presented in this dissertation

for fabricating complex soft robotic structures. **Chapter 3**, *Sensuator*, is a brief discussion about the use of neural networks for soft robotic sensing and state estimation using a soft robot fabricated using Negshell Casting as an example. **Chapter 4** is deep dive into haptics for virtual reality, by introducing *Chasm*, a compact screw-based haptic actuator. Chasm is used as a platform to explore handheld and wearable output devices for virtual reality. Chasm also briefly explores how mechanically grounding devices to various parts of the body changes the perception of the same haptic stimulus. **Chapter 5** introduces *Haplets*, small, wearable fingertip haptic displays for virtual reality that demonstrates the effect of perceived haptic sensations that are off from the intended site. Haplets also serves as a foundation for device design and rapid prototyping for virtual reality. **Chapter 6** is the culmination of efforts from all previous chapters by presenting a wearable electromyograpghy-based (EMG-based) system for virtual reality that can actively sense and correct for loose contact with

the skin. We study how good and poor sensing can affect the performance of EMG-based input modalities in virtual reality. Using the collected data from our study, we built a system that utilizes soft robotic actuators, state estimation and control to actively maintain good EMG electrode contact with the wearer's skin for our wearable device. Lastly, **Chapter 7** summarizes our efforts and and provides insights on how our work lays foundation for the improvement of wearable devices in the near future.

Chapter 2

NEGSHELL CASTING: 3D-PRINTED STRUCTURED AND SACRIFICIAL CORES FOR SOFT ROBOT FABRICATION

Pornthep Preechayasomboon[1], Eric Rombokas[1]

1 Mechanical Engineering, University of Washington

This chapter was originally published June 12, 2020 in PLoS ONE [108]

Abstract

Soft robot fabrication by casting liquid elastomer often requires multiple steps of casting or skillful manual labor. We present a novel soft robotic fabrication technique: negshell casting (negative-space eggshell casting), that reduces the steps required for fabrication by introducing 3D-printed thin-walled cores for use in casting that are meant to be left in place instead of being removed later in the fabrication process. Negshell casting consists of two types of cores: sacrificial cores (negshell cores) and structural cores. Negshell cores are designed to be broken into small pieces that have little effect on the mechanical structure of the soft robot, and can be used for creating fluidic channels and bellows for actuation. Structural cores, on the other hand, are not meant to be broken, and are for increasing the stiffness of soft robotic structures, such as endoskeletons. We describe the design and fabrication concepts for both types of cores and report the mechanical characterization of the cores embedded in silicone rubber specimens. We also present an example use-case of negshell casting for a single joint soft robotic finger, along with an experiment to demonstrate how negshell casting concepts can aid in force transmission. Finally, we present real-world usage of negshell casting in a 6 degree-of-freedom three-finger soft robotic gripper, and a demonstration of the gripper in a robotic pick-and-place task.

A companion website with further details about fabrication (as well as an introduction to molding and casting for those who are unfamiliar with the terms), engineering file downloads, and experimental data is provided at `https://negshell.github.io/`.

2.1 Introduction

The fabrication of soft robotic structures is often a tedious process with multiple steps [125, 61]. Soft robots benefit greatly from complex mechanical structures and geometric features to achieve actuation and sensation. Unfortunately, the soft and compliant nature of the materials used for soft robots, such as silicone or urethane-based elastomer, are only compatible with few fabrication methods such as injection molding and casting. Casting has tremendous benefits in terms of the precision in the external surface of the casted part, which is why it is often used for prosthetics and props for the film production industry, and for end-user products in the plastic molding industry. However, soft robots have structures that span both the external and internal space and most rely on internal features to actuate and sense. For example, the ubiquitous PneuNets [90] actuator relies on expanding internal bladders, while some sensing modalities require fluids embedded within the soft robot [107]. Casting these internal features is not a straightforward process. The fabrication process often requires multiple steps: first, a "core" that will create the internal features must be casted around and removed, then, another section is additionally casted to complete the body or another piece of the body is bonded onto the first part. Because these cores have to be removed during the fabrication process, they either: 1. have geometry constraints, such as minimal undercuts and overhangs, to prevent difficulty of removal, or 2. are made to be dissolved later, also known as lost-wax casting [61]. Lost-wax casting enables more complex geometry for the core, but the core itself must be cast from wax, which is prone to shrinkage and breakage. Aside from wax, 3D-printable dissolvable materials can be used to create arbitrarily-shaped molds and cores, such as polyvinyl alcohol (PVA) which dissolves in hot water or acrylonitrile butadiene styrene (ABS) which dissolves in acetone [69]. However, dissolving the material takes a large amount of time (more than 3 hours, according to [69])

and requires the core to be accessible from the outside of the soft robot to drive liquid solvent through.

3D printing soft robots is another heavily-researched fabrication process. 3D printing enables internal and external features to be fabricated directly and with multiple materials simultaneously. However, some solutions are prohibitively slow for mass manufacturing while others have limitations in geometry due to the absence of support material [149, 154]. Resin-jetting, multi-material printers such as the Objet Connex (Stratasys, Ltd.) series have both support material and elastomeric-like resins, but suffer from poor mechanical properties when compared to typical silicone rubbers [59]. Some stereolithography (SLA) [150, 32], digital light projection (DLP) [156, 97, 136] or continuous liquid interface production (CLIP) printers have high-resolution and fast print times, but can only print with one material at a time, which leads to support structures that need to be printed from the same material and then later removed. Enclosed chamber-like features, such as for bellows and deformation sensors, printed with SLA-style printers also suffer from cupping artefacts which can lead to ruptures [34].

High resolution 3D printing with SLA-like methods does have its benefits, however, in the casting process, as the features in the outer shell molds can be as complex and intricate as the designer desires due to SLA's high-resolution. Furthermore, resin based printing can create thin-walled parts and small cavities for use in microfluidic devices [148]. In this paper, we combine the benefits of casting liquid silicone rubber and SLA-based resin printing by 3D printing thin-walled cores for the internal features of soft robots that are meant to be left in place after casting. This greatly reduces the steps required for fabrication as casting is only done once — since the core is never removed. These thin-walled cores can be used as sacrificial elements such as for creating expanding bellow chambers and fluidic channels, or used as passive structural elements such as fingernails or bony features in soft robotic fingers. Another benefit is that since the the thin-walled cores are devoid of material, they can occupy space in the form of air instead of heavy silicone rubber, which is useful for creating features that require high stiffness and low weight. We characterize these sacrificial

thin-walled cores (negative-space eggshells, or *negshells*) and non sacrificial cores (structural cores) casted in silicone rubber through mechanical testing. We also provide the fabrication steps for 3D printing our cores and molds, and guidelines for designing soft robots that employ our fabrication method along with an example to show the efficacy of the cores when combined. We also present an example application in the form of a three-finger gripper, as shown in Figure 2.1, performing simple pick-and-place tasks while mounted on a robotic arm.

Figure 2.1: **Negshell casting used for a three-finger soft robotic gripper.** Pictured are three stages of negshell casting from the rightmost to the leftmost finger: 1. 3D-printed structural and negshell cores, 2. The cores placed in 3D-printed molds to be injected with silicone rubber, and 3. A completed finger ready to be actuated.

2.2 Design and fabrication concepts

In this section, we describe the design of each component for negshell fabrication. We contextualize and describe sacrificial and structural cores. Next, we describe how these cores interact with the remainder of the mold and how they are affixed into place for casting.

We then present the design of an example application, a bellow-jointed finger incorporating both kinds of cores, and demonstrating how this relatively common design theme can be miniaturized and fabricated faster and more reliably.

2.2.1 Sacrificial cores

Soft robotic features such as expanding bellows or sensor cavities require a void to be left after fabrication that is later filled with fluids. As described previously, these voids are often created by solid geometry (cores) that represents the void that is later removed either by means of dissolving (lost-wax casting) or removed during the casting process. In our fabrication process, we replace these cores with 3D printed thin-walled volumes (negative-space eggshells or negshells) that are meant to be left in place or broken into small pieces that minimally mechanically effect the surrounding structure. The negshell cores are simply the outer surface of the desired core with a thickness of 0.4 mm, which is the minimum supported wall thickness for the Formlabs' resin we use to print them. This minimum thickness is used to maintain the shape of the volume during handling while being fragile enough to break under minimal force. To help promote breakage, a cross-hatch pattern, shown in Figure 2.2, is cut throughout the surface, as shown in Figure 2.3. The patterned minuscule slots also serve as channels for uncured resin to escape during the print process. Finally, 1.5 mm diameter holes are created for support structures for suspending the cores in the mold, which is further explained in the **Outer molds and support columns** section.

We design our negshell cores as parts in SolidWorks, but the process can be adapted to any modern CAD software. We start by modelling the solid representation of the desired core. Then, the core is hollowed out first by copying the entire external surface followed by thickening it to 0.4 mm inwards. Finally, the hollow shell is perforated using the cut-extrude command with the previously described pattern. A brief overview of the design steps in shown in Figure 2.3. The part is then exported as a stereolithography (.STL) file for printing. We then use Formlabs' Preform software to prepare the negshell part for printing. The parts are first oriented for easy support removal and then Preform generates

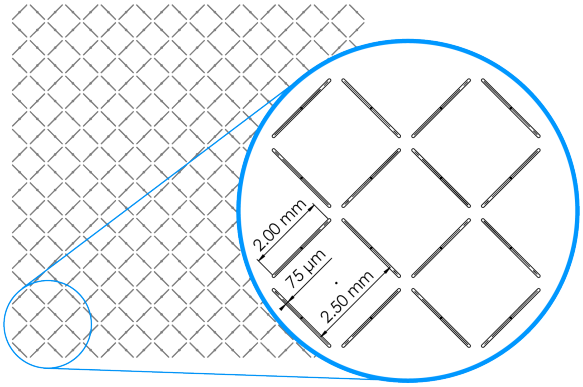

Figure 2.2: **Cross-hatch pattern for perforating negshell cores.** The cross-hatch are 75 µm wide by 2 mm long slots oriented in a 2.5 mm by 2.5 mm square pattern. The slots are designed to be small enough to be partially fused during printing to keep geometric integrity but also become fragile under load.

the necessary support structures for the parts. We chose a support density of 0.5 and a support touch size of 0.4 mm. Our parts were printed using Formlabs' Clear V4 resin with a layer size of 100 µm. Additional details for fabrication, sample parts as well as .form files for Preform are available at our companion website: `https://negshell.github.io/`.

Due to the SLA printing process, the parts will be covered with uncured resin after printing. The residual resin must be removed prior to usage. Typically, parts are cleaned by submerging them in 95% isopropanol (IPA) for a 5 to 15 minutes and scrubbed gently with towels or brushes. However, we found that submerging negshell parts in IPA can cause them to swell and crack prematurely due to the absorption of IPA. Instead, we use a paper towel doused with IPA to wipe away the residual resin and the parts are immediately dried with a dry paper towel. The parts can also be further cured using ultraviolet (UV) light in a UV enclosure to attain higher strength and cure any remaining resin. Finally, the support structures are removed using flush-cutters.

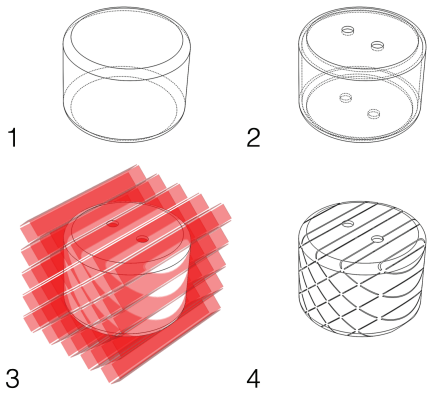

Figure 2.3: **Creating a negshell core in CAD.** 1: The solid model representing the core. 2: The core is hollowed to a 0.4 mm shell. 3: Cutting the cross-hatch pattern into the hollowed core. 4. The final negshell core ready for 3D printing.

2.2.2 Structural cores

Casting is often performed by injecting or pouring a homogeneous material and having it set. Casting multiple materials is possible, but requires a multiple-step process, such as overmolding: casting additional elastomer on top of an existing injection-molded rigid part in a separate mold. Overmolding is often used for electric power tools and cable assemblies to add compliance to an otherwise rigid part. For soft robots, having the entire structure comprised of a single material limits the design space due to the high compliance that is homogeneous throughout. By embedding semi-rigid, passive structures inside the soft robots' body, the localized mechanical properties at those structures can be tuned. We achieve this by utilizing cores similar to negshell cores but do not have the perforations. Such structures provide a semi-rigid internal skeleton that provides a stiffer backing to exert force akin to fingernails in humans, or provide lightweight void space for areas that connect one structure to another that would otherwise be heavy and undesirably non-rigid if made entirely from elastomer.

We design and build these structural cores in the same manner as the negshell cores: a hollow shell with minimum wall thickness. The thickness can be tuned to create varying stiffness in the resulting part. We present a characterization of these parameters in **Characterization** section.

2.2.3 Outer molds and support columns

The cores must be suspended in the outer mold at its intended location during the casting process, as shown in Figure 2.4. It should be noted that this is a limitation of this fabrication method as the structure necessary to suspend the cores prevents the internal features from being completely sealed. However, the support structures can be made small in size and the resulting holes can be sealed with a small amount of adhesive or filler, such as Sil-Poxy (Smooth-On, Inc.).

The support structures (or standoffs) are round 2.5 mm columns that reach from the inner surface of the outer mold up to the core. At the end of the column, there is a smaller column with a beveled tip approximately 1.5 mm in diameter that will penetrate and secure the corresponding hole on the core, as shown in Figure 2.4. Depending on the geometry of the core, the support columns can be placed at multiple locations around the mold and core. To aid in the ease of removal of the outer mold after casting, the columns should be oriented in the same pulling direction when removing the mold, although this design rule can be somewhat relaxed due to the compliance of the elastomer.

The outer mold can be as simple as two halves of a shell that defines the external features of the desired part, or a complex multiple part mold to accommodate undercuts or overhangs and features that would be otherwise impossible to be demolded with a two-part mold. As our focus of this paper is on the internal features, we designed most of our parts to be cast in simple two-part molds. The molds are often split at the central plane of the desired part that creates the least undercuts. In this paper, our molds have the following set of features to aid in fabrication: 1. The seam connecting the two halves of the mold have a lip to help with alignment and prevent liquid elastomer from escaping. 2. There are 3.1 mm holes

intended for M3 screws and nuts along the perimeter of the mold to clamp the two halves together. 3. Instead of pouring liquid elastomer in, we opt for injecting the liquid elastomer using a syringe. We add two Luer lock ports (Figure 2.4) at opposite ends of the mold: one for injecting the liquid elastomer with a syringe, while the other port is used for drawing a vacuum to assist in flow and for increasing pressure to eliminate air bubbles.

Our molds are 1.5 mm thick and are printed using the same Clear V4 resin as the negshell cores. The molds are thoroughly washed with IPA to minimize any residual resin, as uncured resin can inhibit curing of the Platinum-cure silicone that we use.

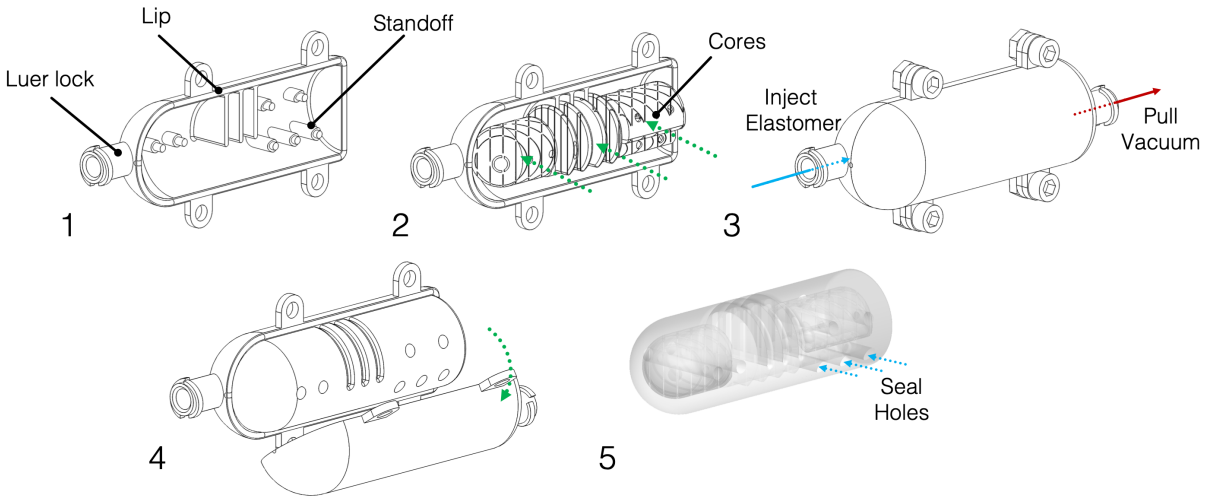

Figure 2.4: **General fabrication steps for negshell casting.** 1: The mold is prepared. The Luer lock, lip and standoff features are highlighted. 2: The cores are inserted into one side of the mold. 3: The mold is clamped shut with fasteners and elastomer is injected via a syringe. Another syringe can be used at the other end to pull a vacuum to help with liquid elastomer flow. 4: Once the elastomer has set, the mold haves can be removed. 5: The completed soft robot component.

2.2.4 Design of a bellow-jointed finger

Here we present an example of a soft robotic finger that utilizes both the structural cores and negshell cores. The finger has multiple bellows connected to an inlet tube that is the result of a negshell core. The tip and base have stiffening and weight reducing elements created by structural cores, as shown in Figure 2.5. The completed finger is a cylinder approximately 18 mm in diameter and 55 mm in length with a domed tip and weighs approximately 15 grams. The bellow negshell core and accompanying mold features create 1 mm wide internal bellows with 1 mm thin walls. The structural cores are located approximately 2 to 3 mm deep from the surface of the finger.

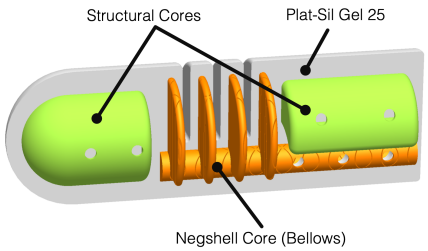

Figure 2.5: **Cross-section of soft robotic finger.** A 3D CAD model of the soft robotic finger is cut longitudinally to show the locations of the non-sacrifical cores and the negshell core that create the bellow

To fabricate the finger, the cores and molds were first 3D printed from Clear resin with the same parameters as mentioned previously. Once the mold and cores have been cleaned from resin residue, the mold's inner surface was lightly brushed with Pol-Ease 2500 (Polytek Development Corp.) release agent and the cores were placed in one half of the mold. The second half was then clamped shut with M3 screws and nuts. 15 grams of liquid silicone Plat-Sil Gel 25 (Polytek Development Corp.) was prepared by vigorously mixing equal amounts of part A and part B which was followed by vacuum degassing to remove trapped gasses, as

we found that air bubbles can lead to leakages near the bellows due to the low thickness. The liquid silicone was then poured into a 30 mL syringe and the silicone was injected into the mold via the Luer lock port at the tip. The Luer lock port at the other end serves as a vent but can also be used with another empty syringe to draw a vacuum to aid in the flow of liquid silicone. Both syringes were left in place during curing to prevent silicone from flowing out of the mold. After waiting for the silicone to cure, which is approximately one hour for Plat-Sil Gel 25, the syringes was removed and the mold was opened. Small holes created by the standoffs were sealed with Sil-Poxy (Smooth-on, Inc.) silicone adhesive. The same adhesive was used to bond a 4 mm silicone tube to the inlet. Finally the negshell cores were crushed by hand, completing the finger.

During operation, as pressure is increased in the cavity created by the negshell cores, the bellows expand and push against each other which causes the finger to bend. Since the silicone and 3D printed cores do not form a bond, the fluid used to actuate the finger can travel along the surface of the bellow fins and gaps left after crushing the negshell core walls. The bellows also expand against the stiff structural cores, creating a clear path for force transmission from the tip of the finger to the base of the fingers.

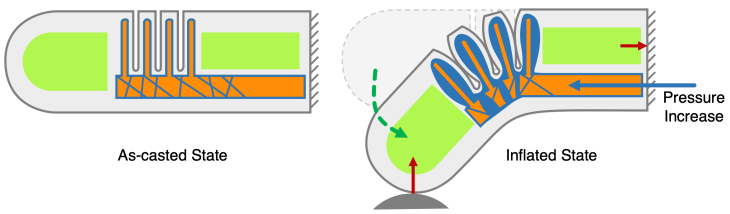

Figure 2.6: **Main actuation principle of the soft robotic finger.** As the actuation fluid (air/water) increases in pressure or volume, the fluid travels along the cavity left by the negshell core up to the fins of the bellows and start to expand the thin walls of the silicone elastomer surrounding the fins which creates a bending force.

2.3 Characterization

To evaluate the efficacy of the cores, the following section presents several mechanical characterizations of both negshell and structural cores. We characterize: 1. the force required to break the negshell cores, 2. the stiffness of elastomer with negshell cores post-break, 3. the stiffness of elastomer with structural cores, and 4. a comparison of the blocked force output of bellow-jointed fingers with and without structural cores.

2.3.1 Specimens and experimental setup

Specimens for both types of cores are hollow cylinders with a diameter of 16 mm and a height of 11 mm. The cores were then casted into a cylinder 20 mm in diameter and 15 mm in height with Plat-Sil Gel 25 Platinum-cure silicone, which resulted in an elastomer thickness of 2 mm across the whole specimen. The casting process is similar to the processes detailed previously. We fabricated three samples of each type of core, including comparable samples that can be achieved with traditional casting methods, as shown in Figure 2.8 and Figure 2.9. All samples were tested on a TA ElectroForce 3200 Mechanical Tester (TA Instruments) fitted with a 50 mm platen and 120 N capacity load cell.

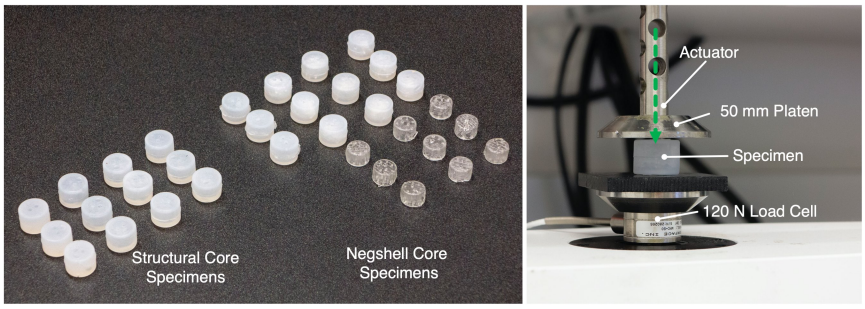

Figure 2.7: **Experimental setup.** (Left) All of the specimens used for testing. (Right) The specimen is placed at the center of the platen. As the actuator moves the platen down, the specimen is compressed and the resulting force is read from the load cell beneath.

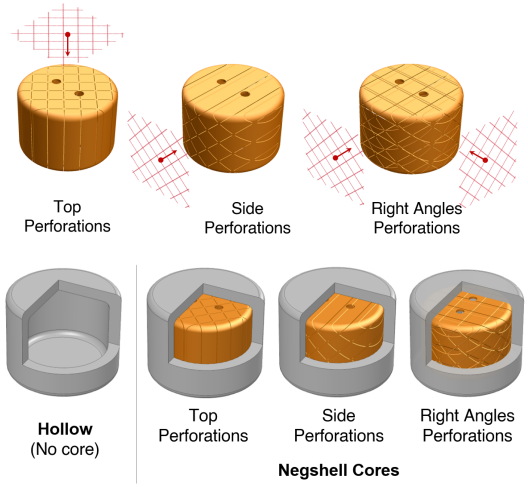

Figure 2.8: **Negshell core specimens.** The four types of negshell core silicone specimens tested. A zonal cross-section shows the internal features of the specimens. The hollow specimen does not contain any core. The negshell core specimens contain the three different cores with different perforation directions.

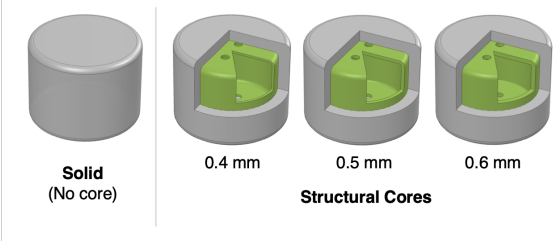

Figure 2.9: **Structural core specimens.** The four types of strutural core silicone specimens tested. The solid specimen has homogeneous material throughout its volume. The cores in the other specimens have a thickness of 0.4 mm, 0.5 mm, and 0.6 mm.

Each specimen was also weighed. The average weight of each specimen type from Figure 2.10 shows that for negshell specimens, the negshell cores contribute approximately 27% of

additional weight and the structural cores reduce weight by up to 67% for these specimens.

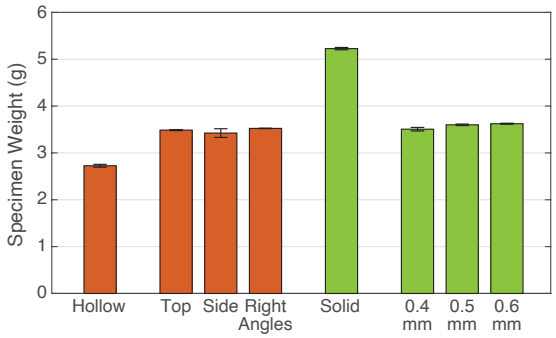

Figure 2.10: **Specimen weights.** Three samples of each type of specimen was weighed on a 0.01 gram resolution scale and averaged. The errorbars represent the standard deviation.

2.3.2 Characterization 1: force required to break negshell cores

As the negshell cores are intended to break into small pieces, we characterize the amount of force required to break the negshell cores. The characterization is done by placing the negshell cores on its side and subjecting the cores to a compression load on the TA Electroforce 2300. The specimens were preloaded to deform slightly, compressed 4 mm further at 0.25 mm/s, and then released at 1 mm/s. We tested three different specimen types with the same wall thickness of 0.4 mm: 1. cores with perforations cut from the circular base of the cylinder (top), 2. cores with perforations cut from one side, and 3. cores with perforations cut at the two sides at right angles, as shown in Figure 2.8. The results show that the cores start to break from 20 - 35 N of force and had a wide range of variance between each sample, as shown in Figure 2.11. This is, however, well within range of the force a human can exert with a pinch grasp.

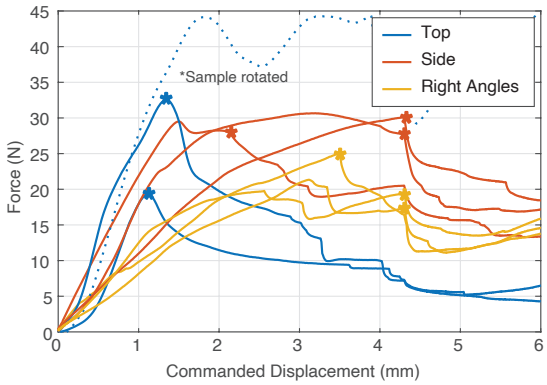

Figure 2.11: **Breaking force for negshell specimens.** Each specimen required different amounts of force to start rupturing, as denoted by the asterisks that precedes a sharp decline in force reading.

2.3.3 Characterization 2: stiffness of elastomer with negshell cores post-break

Negshell cores, once broken, are intended to minimally affect the mechanical properties of the surrounding elastomer and provide the same function as a removed core. To demonstrate this, we cast three different types of negshell specimens (top perforation, side perforation and perforation at right angles) and compare them to specimens with the equivalent resulting geometry that were fabricated with a traditional casting process: casting with a solid core in halves and then bonded together. Each specimen was first subjected to a preload of 1 mm of deformation followed by a 0.5 Hz triangle wave of compression with a peak-to-peak displacement of 4 mm for five cycles. The average stress-strain curve of each type of negshell specimen is shown in Figure 2.12. The stress in the specimen is calculated by dividing the force read from the load cell by the area of the base of the specimen. From the stress-strain curve, it can be implied that the Top and Right Angles specimens behave similarly to the specimens without the cores. The Side specimens have slightly higher overall stiffness, as shown in stress-strain curve and the resulting elastic modulus. The average elastic modulus

(E) (derived from an approximate linear fit of the stress-strain curve) of the different core types that is shown in Figure 2.12 confirms that the the broken negshell cores contribute an insignificant amount of stiffness to the casted samples.

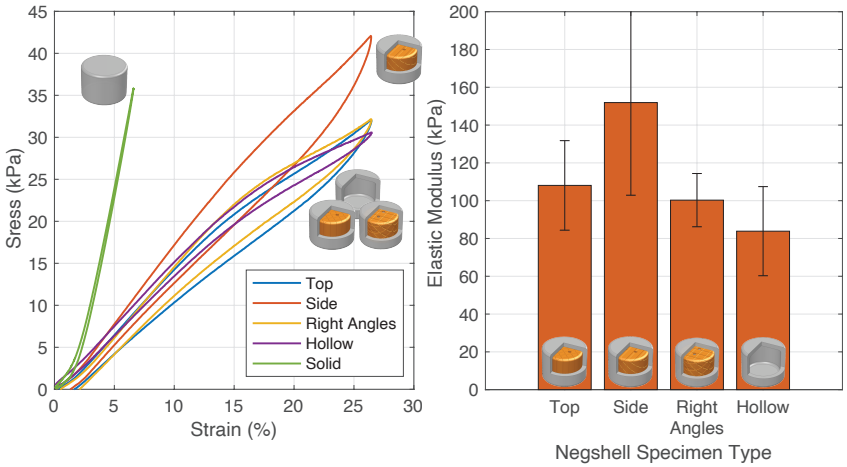

Figure 2.12: **Negshell cores characterization results.** (Left) The stress-strain curve of the tested specimens show that the negshell cores minially effect the mechanical property of the specimens under compression. The stress-strain curve of a solid, homogeneous specimen is also plotted as refrence. (Right) The elastic modulus of the tested specimens.

2.3.4 Characterization 3: stiffness of elastomer with structural cores

To demonstrate that structural cores with thin walls can modulate the overall stiffness of soft robotic elements, we casted specimens with structural cores with three different thicknesses: 0.4 mm, 0.5 mm and 0.6 mm and compare them to a equivalent homogeneous specimen without a core. Each specimen was subjected to a 0.5 mm displacement preload followed by a 1 mm peak-to-peak displacement triangle wave at 0.5 Hz for five cycles. We did not exceed this displacement range due to the amount of force reaching the upper limits of the load cell in some samples. The average stress-strain curve of each type of specimen and accompanying

elastic modulus is shown in Figure 2.13. It can be seen from the resulting elastic moduli that the stiffness of the overall structure can be increased by over 500%, while reducing the weight by 67% (Figure 2.10).

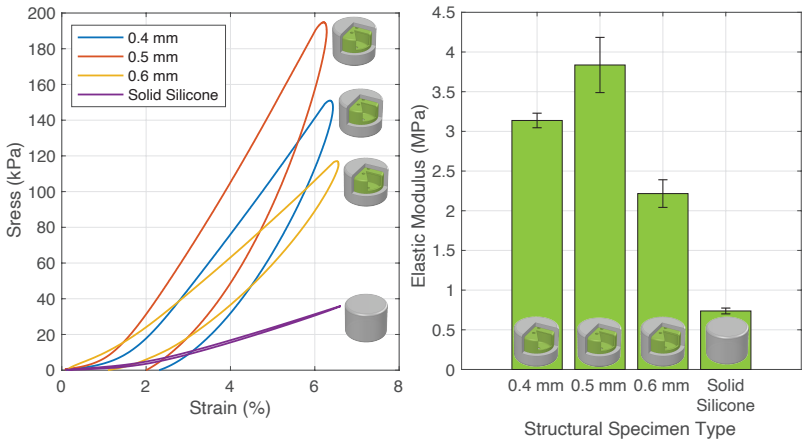

Figure 2.13: **Structural core characterization results.** (Left) The stress-strain curve of the tested specimens show that the stiffness of the specimens can be greatly increased with the structural cores. (Right) The elastic modulus of the different structural cores.

2.3.5 Characterization 4: a comparison of the blocked force output of bellow-jointed fingers with and without structural cores

As previously hypothesized (Figure 2.6), the structural cores help provide a means of force transmission from the tip of the finger to the base. We test this by comparing two fingers fabricated with and without structural cores in a blocked force experiment. The finger without the structural core was fabricated by simply not inserting the structural cores into the mold. The fingers were secured with a cable tie to a rigid aluminum structure at the base and a load cell is placed under the finger approximately at the location where the finger tip touches at 15 degrees of flexion. The finger is manually pressurized using a 30 mL syringe

connected through a silicone tube to the bellow structure.

The results from the blocked force experiment, in Figure 2.14, show that the finger with the structural cores can exert a maximum force of 1.83 N with 64 kPa of air pressure while the finger without the cores can only provide 1.52 N of force with the same pressure — a 20% increase in force output with the structural cores. Furthermore, the structural cores reduce the overall weight of the finger by 1.8 grams or 14%.

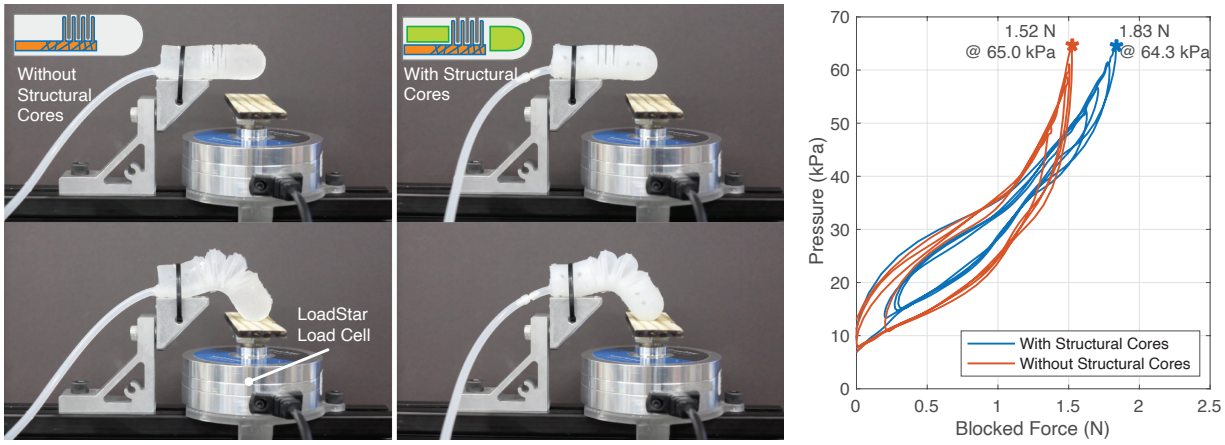

Figure 2.14: **Blocking force output of the bellow-jointed fingers.** (Left) The experimental setup. The fingers in the top row are in the idle state the bottom rows are at full flexion. (Right) A plot showing the relation between input pressure and the amount of force exerted by the finger.

2.4 Application

In this section, we combine the design elements of negshell casting in an example application where we fabricate a three-finger gripper with 2 bellow-joints per finger. The gripper demonstrates the use of negshell cores for creating expanding bellows at each joint and their fluid channels, and structural cores to aid in force transmission, local stiffness, and to secure the finger to the base. We also mount the gripper on the tool plate of a Barrett WAM 7-

degree-of-freedom robotic arm (Barrett Technology, LLC) to demonstrate a simple real-world scenario of a pick-and-place task.

2.4.1 Design and fabrication

Each phalanx of the gripper can be independently actuated by applying fluid pressure on each of the two bellow-joints: the proximal joint and the distal joint. The proximal joint connects the structural core in the base to the middle structural core and the distal joint connects the middle structural core to the finger tip, as shown in Figure 2.15. Each joint consists of four fan shaped 0.6 mm thin chambers with a diameter of approximately 16 mm, similar to the bellow-jointed finger in the previous section. A 3 mm tubular negshell core connects the bellows to the exterior of the finger through two holes in the base — one for each joint, which is later used for inserting and bonding a 3 mm silicone tubes to connect the chambers to a pressure source. The middle structural core is a half-cylinder with a thickness of 0.6 mm. The core at the fingertip is a hybrid between negshell and structural cores, where the "pulp" of the fingertip core is to become soft and compliant to emulate soft human fingertip tissue while the immediate structural core connected to the negshell core emulates the hard backing provided by the bony structure and fingernails in humans, as shown in the inset in Figure 2.15. We intend for the fingertip to become a sensor, but it is beyond the scope of this paper. The mold is similar to the mold presented in the previous sections with one difference where the base is also used as part of the mold. Geometry similar to a structural core is integrated into the base which is used to retain the finger on the base, as shown in Figure 2.15.

We fabricated the gripper by first 3D printing the cores, outer molds and base using Clear resin on a Formlabs Form 3 printer. The parts were then cleaned and processed in the same manner as in the previous sections. The molds were then prepared by applying a light coat Pol-Ease 2500 to aid in demolding. One half of the mold was then assembled onto the base followed by the insertion of the various cores. The other half of the mold was then carefully closed and several M3 screws and nuts clamped the two halves shut. Approximately

30 grams of Plat-Sil Gel 25 was mixed, vacuum degassed and loaded into a syringe, then the liquid elastomer was injected into the port at the fingertip. Another syringe was used at the base to draw a vacuum or increase pressure to help with residual air bubbles. Once the elastomer set, the two halves were released and Sil-Poxy was used to seal the holes created by the standoffs, and to bond silicone tubes to the fluid channels for the bellows and fingertip. Each finger can be done one at a time, or all three at once if three sets of molds were printed. Finally, each negshell core is broken by hand and the gripper is ready to be mounted to the WAM Arm tool plate using M6 hexagon cap screws.

2.4.2 Actuation

There are a total of six bellow-joints in the gripper, which can be individually actuated by modulating the pressure within the bellows of each joint. Pressure modulation can be done in multiple ways such as using multiple pulse-width modulated (PWM) valves and a pressure source [161] or multiple leadscrew driven syringes [59]. However, as robust actuation and dexterous manipulation is beyond the scope of this paper, we simply combined all the fluid channels together into a single channel so all joints are pressurized with a single pressure source, which in this case is a hand-actuated syringe. However, individual actuation of each bellow using a hand-driven syringe is shown in Figure 2.16.

2.4.3 Pick and place demonstration

As a demonstration of real-world usage of the negshell casted three-finger gripper, we performed a brief demonstration by using the gripper mounted on a Barrett WAM Arm to perform pick and place tasks of everyday items. We pre-recorded the arm movement using a "Teach and Play" program that is then played back each time the arm performs a pick and place task. The arm starts from its home position, moves to position the gripper directly above the object to grasp, and moves the gripper down to encompass the object with the fingers. The fingers are then closed using the syringe and the arm moves the object with the gripper towards a bin. Finally, the fingers are opened and the object is dropped into the bin.

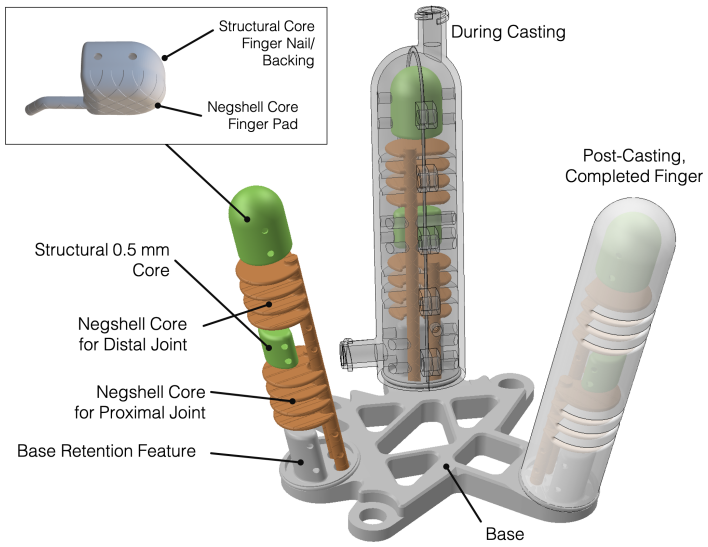

Figure 2.15: **Overview of three-finger gripper's fabrication.** The 3D-printed base, which serves as a part of the mold for the fingers, provides a rigid connection from the fingers to the tool plate of the WAM Arm. Two seperate negshell cores are used for each joint of the fingers. A structural core aids in rigidity and force transmission between the joints. The fingertip contains a hybrid structural-negshell core with renders a soft finger pad and solid backing, as shown in the inset. The base also contains a retention feature geometrically similar to the structural cores, where silicone will be mechanically trapped and locked in place due to the overhang.

We tested grasping a total of 28 objects varying in shape, size and density, with most shown in Figure 2.17 e.g. a water bottle, a lightbulb, and a 5 mm hex key. Some objects were handed to the robot, such as the water bottle and pen, while others were placed on a pedestal to be picked up, such as the wooden block and cloth. It can be observed from the footage that the gripper successfully conforms to objects with arbitrary geometries and poses despite the bellows being actuated all at once, demonstrating proper compliance as expected from a soft robotic gripper.

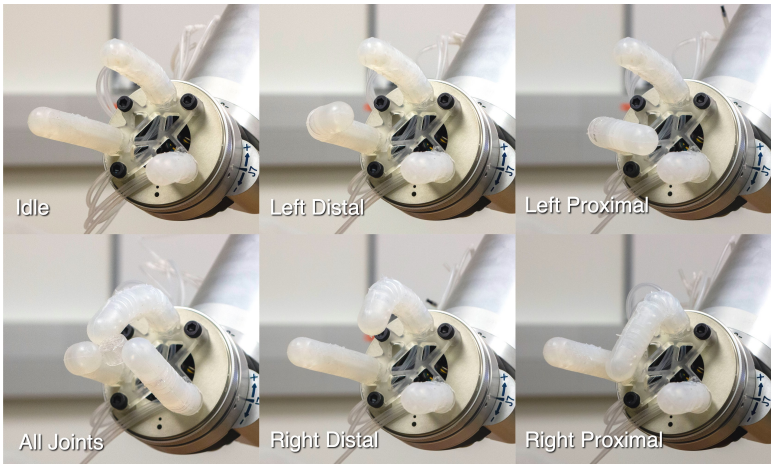

Figure 2.16: **Pressurization of each joint of the gripper.** (Top-leftmost) The gripper in its un-pressurized, idle state. (Bottom-leftmost) All joints being pressurized at once. (Right four photographs) A series of photographs showing each joint of two of the gripper's fingers pressurized.

2.5 Discussion

Negshell casting opens up new possibilities for casted soft robotic designs by integrating 3D-printed parts both into the structure and the casting process — cutting down the casting process while increasing the design space significantly. The internal geometry of soft robotic structures can be much more complex than what could previously be done through casting. However, negshell casting does have limitations. Since the cores must be suspended during casting, some support structure must exist, which requires sealing the holes created by the structures. Though straightforward, sealing the holes is an additional step that is done manually and prone to human error. In the future, we hope to address this step to further streamline the fabrication process for the sake of scalability and manufacturability.

Sensing, although mentioned in the previous sections, has not been demonstrated in this paper. However, the mechanical characterization of the negshell specimens show that, under

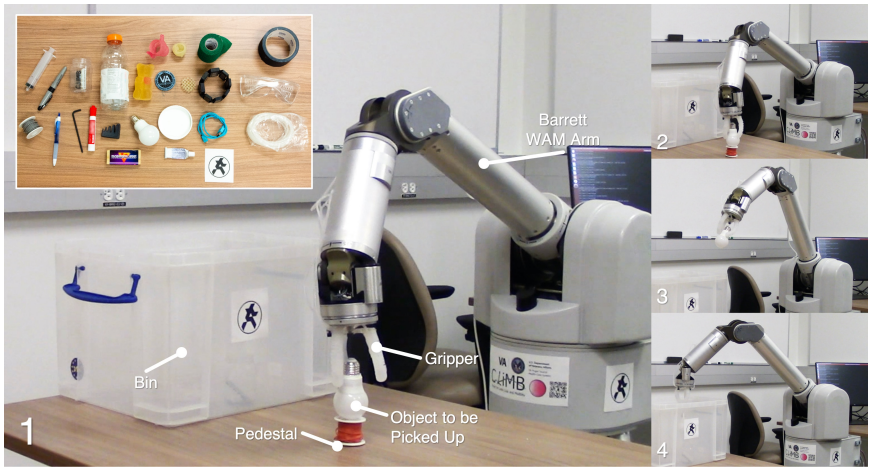

Figure 2.17: **Pick and place demonstration.** The three-finger gripper is shown attached to a Barrett WAM Arm. The arm goes through a pre-recorded motion while the gripper is manually actuated with a syringe (not shown). 1. The arm positions the gripper to encompass the object to be picked up, 2. All bellows in the gripper are pressurized at once using a syringe, grasping the object, 3. The arm moves the object up over the bin, 4. The object is dropped into the bin upon the release of pressure. Inset: the objects that were successfully picked and placed.

compression, the broken negshell cores have little to no effect on the structure, thus negshell casting could possibly be used for creating sensor channels or structures. An example of this is shown in the fingertip of our three-finger gripper, where the fingerpad is deformable chamber that can be connected to a pressure sensor for simple tactile sensing. Sensing will "close the loop" for our soft robotic grippers and thus we will explore sensing and closed-loop control in the near future.

3D-printing soft robots, especially with complex internal structures, is still an active research topic. Work such as [143], still requires molds and manual labor within each layer of printing. The most notable hurdle is arguably the material science required to develop the ideal printing material. Our technique leverages already mature materials for 3D-printing

and adapted it to a tried and true method for fabrication. For future work, combining the two technologies may aid in each-other's shortcomings. For instance, 3D-printing can be done on top of negshells and structural cores that are used for scaffolding which eliminates the need for sealing holes that are created during casting, while providing instant negative space that must be otherwise printed with fugitive materials.

One concern that may be raised is that the fragments that are created during fabrication may cause blockages or damage to the structure. To mitigate blockages, the design of the cross-hatch pattern may be tweaked in a manner where the distance of the perforations are smaller than the smallest channel that is exposed to the negshell core. In practice, however, we hypothesize that due to the inherent compliance of the structure, fluids would simply flow around the blockage. Furthermore, the fragments themselves are flexible and are compliant themselves due to the 0.4 mm thickness. As for damage, as crushing and breaking the negshell cores is part of the manufacturing process, the soft robotic structure has already been subjected to forces much higher than typical usage conditions. Our combination of materials shown in this paper has proved to be able to withstand the process. As for other materials, the parameters of the negshell core might require tweaking, i.e. using a thinner wall thickness or larger perforations for silicone with less hardness.

2.6 Conclusion

We have presented a novel fabrication technique, negshell casting, that leverages SLA 3D-printing to create complex cores for use during an otherwise traditional casting processing. We present two types of cores: negshell cores and structural cores. Negshell cores are sacrificial cores that are used to create soft or flexible internal structures such as channels and bellows while structural cores are used for modulating stiffness and for weight reduction. We showed that the negshell cores have little effect on the mechanical structure of silicone elastomer and structural cores can increase the stiffness of silicone structures significantly. Our demonstration of the bellow-jointed finger and the three-finger gripper shows that negshell casting can be used for creating soft robotic actuators. In the near future, we will explore

how negshell casting can enable more sophisticated and integrated sensing capabilities in soft robotic designs.

Chapter 3

SENSUATOR: A HYBRID SENSOR-ACTUATOR APPROACH TO SOFT ROBOTIC PROPRIOCEPTION USING RECURRENT NEURAL NETWORKS

Pornthep Preechayasomboon[1], Eric Rombokas[1]

1 Mechanical Engineering, University of Washington

This chapter was originally published February 7, 2021 in Actuators, MDPI [110]

Abstract

Soft robotic actuators are now being used in practical applications; however, they are often limited to open-loop control that relies on the inherent compliance of the actuator. Achieving human-like manipulation and grasping with soft robotic actuators requires at least some form of sensing, which often comes at the cost of complex fabrication and purposefully built sensor structures. In this paper, we utilize the actuating fluid itself as a sensing medium to achieve high-fidelity proprioception in a soft actuator. As our sensors are somewhat unstructured, their readings are difficult to interpret using linear models. We therefore present a proof of concept of a method for deriving the pose of the soft actuator using recurrent neural networks. We present the experimental setup and our learned state estimator to show that our method is viable for achieving proprioception and is also robust to common sensor failures.

3.1 Introduction

Progress in soft robotics research is accelerating, and soft robots are impacting a variety of practical applications, from grasping to search-and-rescue. The steady advances in this

area have generally focused on integrating physical intelligence for robust and useful open-loop behavior [80, 90]. By taking advantage of characteristics such as compliance, passive stability and flexibility, the advantages of these robots have mostly been realized using only limited sensing. We expect continued progress in this domain, but soft sensors are a key capability for realizing the full potential of soft robots. Soft sensors are challenging to design and fabricate, and it can be difficult to interpret their outputs. Despite these challenges, significant progress has been made using a variety of sensor modalities including piezoelectric tactile sensing [123, 76], resistive [95, 85], capacitive [35] and inductive [151] stretch sensors, magnetic effects [44] and optical waveguides [140, 159, 133]. Integrating soft sensors relies on more than taking advantage of various modalities; the sensors must be physically coupled to the robot in useful ways, and the data must be useful for planning or control.

Ideally, we could design and fabricate highly sensorized soft robots to rival the sensor density and acuity found in the animal world. However, dedicated sensor structures add fabrication complexity and require physical space. Biologically inspired sensor strategies would ideally include dense, diverse sensors in currently unrealizable quantities. For instance, the fingertip contains an estimated two to four sensory organs for sensing touch per square millimeter [62]. Multimaterial 3D printing has enabled the fabrication of embedded sensors using conductive ink for touch, pressure, motion and temperature sensing [143, 142], which is somewhat similar to the multi-modal nature of biological systems. Recent work, drawing inspiration from the human perceptual system, has demonstrated that a redundant, unstructured sensor topology may be modeled using modern neural networks. This approach enabled force and displacement models for a soft robot that were robust to sensor non-linearities and drift [138]. To date, the aforementioned approaches have relied on purposefully built sensor structures within the soft robot and may require the use of machine learning to fully utilize the sensor responses. In a similar vein, with the non-linear behavior surrounding the materials and unstructured nature of soft robotics, various facets of soft robotics research have turned to the use of data-driven or deep learning based methods for tasks such as design and fabrication [15], sensing [65, 121], state estimation [141, 126] and control [49, 137, 160].

In this proof-of-concept paper, we present a hybrid sensor–actuator design that, instead of relying on purposefully built sensor structures, takes advantage of the actuated fluidic medium of the robot to perform a double duty as a sensing medium in a bellows-style soft robotic actuator. Prior work by Helps and Rossiter [45] has demonstrated the use of salt water as both an actuated and sensing medium to derive the bending angle of a bellows-style actuator and closed-loop control. That system used a single sensor to estimate the whole-robot bending angle. In contrast, our approach relies on a distributed network of sensors that are created by the volume and geometry of salt water in each bellow. We also use flexible conductive silicone electrode segments that more closely match the material properties of the robot. Due to the unstructured nature of our sensors, we rely on a data-driven model, a recurrent neural network, for proprioceptive state estimation. The multi-channel sensor-actuator provides a six-dimensional space that captures both the tip location and high-fidelity pose. We present the design, experimental setup and state estimation examples of the sensor–actuator hybrid as well as a brief investigation into common sensor failure modes and an exploratory machine learning architecture investigation for our data.

3.2 Materials and Methods

In this section, we provide details on how we fabricated our hybrid sensor–actuator, as well as an experimental setup for validating our approach. Our soft robotic actuator, as shown in Figure 3.1, was a bellows-style actuator similar to the PneuNet actuator [90], where, upon an increase of pressure inside the bellows, the actuator would bend. Each bellow had a conductive silicone tab on each side that could conduct electricity from the internal fluid to the outside; thus, each bellow formed a resistor when a conductive fluid was used. There were seven bellows in total, and the actuator was approximately 25 mm by 26 mm by 90 mm in size. To track the actuator's position in space, we attached seven infrared (850 nm wavelength) LEDs underneath each bellow.

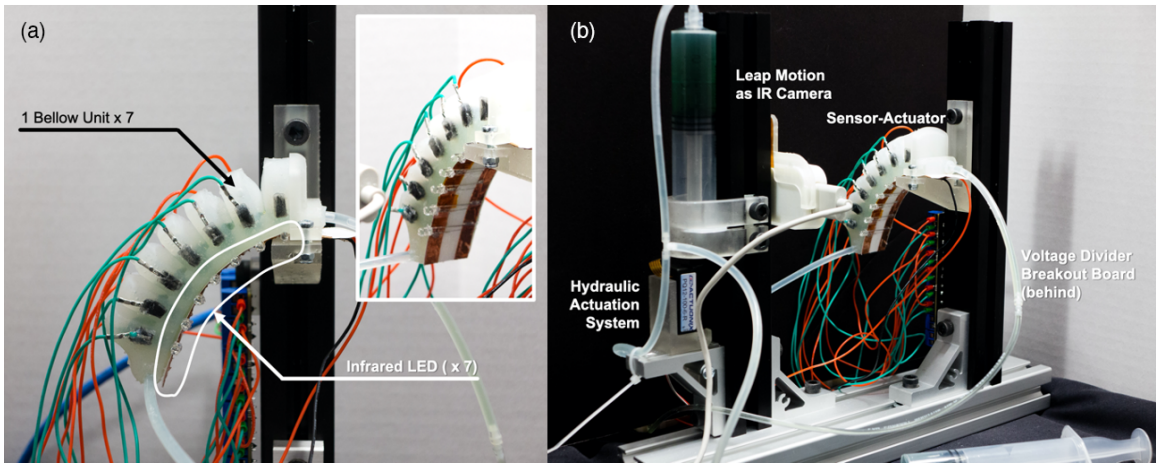

Figure 3.1: (**a**): Our soft robotic sensor–actuator in its idle pose. Shown in the inset are the seven infrared LEDs on a copper strip attached to the actuator, which also acts as a strain-limiting layer. (**b**): The experimental setup with the hydraulic actuation system is shown, along with the Leap Motion infrared camera and breakout board for data acquisition.

3.2.1 Actuator Fabrication

Our bellows sensor–actuator hybrid was fabricated primarily using the Negshell casting method [108]. Negshell casting is a sacrificial casting method that utilizes the inherent fragility of stereolithography-printed (SLA) resins to create breakable cores that become hollow structures inside soft robots after casting. We first 3D printed the internal sacrificial cores and external molds for our actuator using clear resin on a Formlabs Form 2 SLA printer. The molds had 8 by 4 mm openings along the side of each bellow chamber that could be removed after casting—these openings were later used to house conductive silicone. Then, we prepared 60 g of PlatSil Gel-25 silicone elastomer (Polytek Development Corp., Easton, PA, USA) by mixing equal parts of Part A and Part B, followed by vacuum degassing. The mixture was injected into the mold with a syringe and left to cure for 12 h. After the silicone elastomer had set, conductive silicone, as explained in [107], was casted into the openings on the side of each bellow. After the conductive silicone had set, the sacrificial Negshell

cores were crushed by hand to create flexible inner chambers. Then, 3 mm diameter flexible silicone tubes were attached using Sil-Poxy (Smooth-On, Inc., Macungie, PA, USA)) and any leaks were patched using the same adhesive. Lastly, 28 AWG silicone-sheathed electrical wires were pierced into the conductive silicone tabs to create an electrical connection. The actuator was filled with a 5 wt% sodium chloride (table salt) solution mixed with green food coloring.

3.2.2 Experimental Setup

We built an apparatus to actuate the sensor-actuator using aluminum extrusions as a frame, as shown in Fig. 3.1. The sensor-actuator's base was secured onto a 3D-printed pedestal attached to the vertical extrusion. A syringe pump constructed from a linear actuator servo (PQ12-R, Actuonix Motion Devices Inc.), a 30 mL syringe, and 3D-printed parts was attached and connected to the actuator with silicone tubing. A hand-actuated 30 mL syringe was also hydraulically connected in parallel to the servo-actuated syringe. To track the positions of the infrared LEDs, we attached a Leap Motion Controller (Ultraleap), used as an infrared camera with a global shutter, opposite from the actuator. A silicone tube from the actuator was also connected to a pressure transducer (ABPMANT100PG2A3, Honeywell).

To measure the conductance of each bellow, we used the method outlined in [107], where a sinusoidal waveform of 2 volt peak-to-peak centered at 0 volts and a frequency of 100 Hz was applied across each bellow and the voltage drop across the resistors was measured, as shown in Figure 3.2. We used resistors with a resistance of 56 kΩ as the fixed resistor for each bellow. As the shape of the bellows changed, the resistance across the bellows would also change, which can be loosely modeled using the law of resistance ($R = \rho\frac{L}{A}$). An NI USB-6251 (National Instruments) data acquisition system was used to generate the waveform as well as to measure the voltage across each bellow.

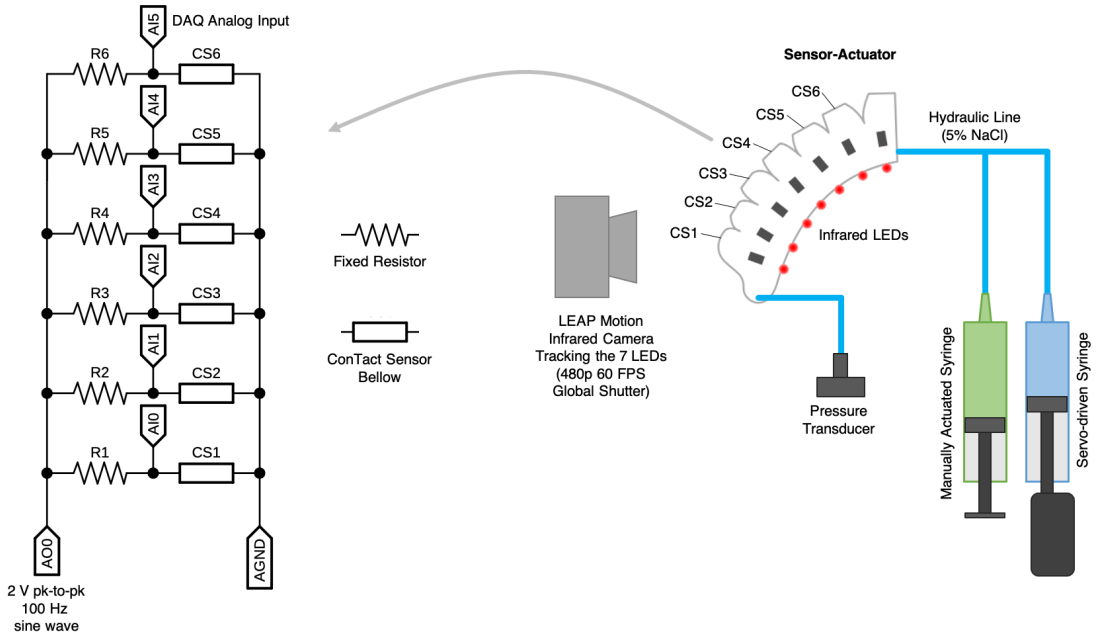

Figure 3.2: Soft hydraulic robotic finger system diagram using hybrid sensor-actuators. Each of the sensing elements (CS1-CS6) spans a bellow of the hydraulic bellows-style actuator. A series of infrared LEDs provides an external "ground truth" of robot pose for training and performance quantification.

3.2.3 Data Collection

The entire aforementioned system was controlled using a PC running LabVIEW (National Instruments) and a Python script. Our Python script processed the image from the left camera of the Leap Motion Controller using the OpenCV library to segment and track the positions of the seven infrared LEDs at 50 Hz. Our LabVIEW application collected the voltage measurements across each bellow along with pressure. The syringe driver's linear servo was controlled using pulse-width modulation (PWM) as the commanded stroke length, which was randomly generated every 1 s. The collected pressure and voltages were sent to the Python script via the local network to be logged to a comma-separated values (CSV)

file, which was timestamped by each camera frame.

We collected two sessions of "babbling", where the actuator was randomly set to different pressures using both the linear servo and a hand-actuated syringe. Throughout each session, we used a hand to block or disturb the actuator at random. The first session lasted approximately 12.5 min or 37,336 samples and served as the training dataset for our neural network. The second session lasted 10 min or 30,000 samples and served as the test dataset for our neural network. A sample video during data collection is provided in the Supplementary Materials.

3.2.4 Machine Learning Models

With the popularity of deep learning and research surrounding the topic, deep learning has transitioned from being a research topic to becoming a tool to enable novel methods in other research areas. In our case, our sensor–actuator's sensor data had several unwanted properties that would lead classical control schemes or models to fail, the most prominent being the non-linearity and coupled interaction between the resistance-based sensors connected electrically in parallel. Thus, using a linear model to derive the actuator's pose from the sensor data was not an option. We therefore turned to deep learning approaches. We chose a recurrent neural network, the bidirectional variant of Long Short-Term Memory (BiLSTM), for our approach. As previously mentioned, research aiming to provide proprioception for soft robots has used recurrent neural networks [138, 141]. LSTM-based models are well suited for time-series predictions and excel in situations where the data are noisy, have time-lagged dependencies and hysteresis [47]. The bidirectional variant of LSTM is fed both data from the past and future during training; thus, BiLSTM is thought to be able to predict future states with higher accuracy due to the better context. We used MATLAB's Deep Learning Toolbox for training and testing our deep learning model, and the relevant hyperparameters are presented in Table 3.1.

MATLAB's Deep Learning Toolbox consists of a set of functions for creating, training, evaluating and using deep learning models. To construct the neural network for our

BiLSTM network, we created our input layer using `sequenceInputLayer()`, followed by a `bilstmLayer()` layer for our BiLSTM layer, and finally a `fullyConnectedLayer()` layer for our fully connected layer. Other architectures could be constructed in the same fashion. A graphical tool, the Deep Network Designer, can also be used to export networks as `.mat` files to be loaded as a single `struct` type variable. After loading our data into memory and a performing pre-processing, we used the function `trainNetwork()` to train our model. Finally, to use the trained model for prediction, we used the `predictAndUpdateState()` function. A MATLAB LiveScript providing step-by-step instructions to train and evaluate our model is available in a repository linked in the Supplementary Materials section. Our raw data is also provided in the same repository.

Table 3.1: BiLSTM Model Properties and Hyperparameters

Parameter	**Value**	**Notes**
Inputs	7	1 × Pressure, 6 × ConTact (normalized)
Outputs	14	X, Y location of 7 infrared LEDs (normalized)
Number of Hidden Units	100	-
Mini-batch Size	512	-
Initial Learning Rate	0.01	-
Learning Rate Drop Period	5	Drop by factor every 5 iterations
Learning Rate Drop Factor	0.5	Next iteration learning rate = 0.5 × previous
Optimizer	ADAM	-
Gradient Clipping	10	-
Iterations	50	-
Final Training RMSE	4.44	in millimeters across all LEDs
Final Test RMSE	5.85	in millimeters across all LEDs

3.3 Results

In this section, we show how our hybrid sensor–actuator can provide rich and informative signals that can be interpreted by learned models. The models were capable of creating reasonable estimates of the positions of the LEDs given time series of sensor data. After training our BiLSTM model, we used it to create time series predictions using our test data as the input. The test data were z-score normalized using the mean and variance for each input from the training data. We describe the preferred estimator's performance in achieving state estimation, examine how sensor degradation affected that performance and compare how different machine learning model choices behaved under sensor degradation.

3.3.1 State Estimation

Over our entire test dataset, which spanned 30,000 samples or 10 min, we achieved a root mean square error (RMSE) of 5.85 mm considering the positions of all seven infrared LEDs. Figure 3.3 shows the entire test dataset along with several snapshots of the actuator's pose. For simplicity, only the position of the fingertip is plotted, ranging approximately 60 mm in the horizontal (X) direction and 40 mm in the vertical (Y) direction. We observe that the model captured the overall pose of the actuator given sensor inputs throughout the time shown. By inspecting the pose snapshots (Figure 3.3 below), we see that at moments where the actuator's tip diverged from the ground truth, the overall pose was still mostly correct. The model also responded well to rapid changes, shown around the 100 to 200 s mark. This result shows that the BiLSTM model generally captured the actuator's nonlinear response and time-dependent characteristics well. Videos of the resulting poses are available in the Supplementary Materials section.

3.3.2 Sensor Degradation

In practice, sensors can degrade over time or fail completely. In biological systems, it is possible to compensate for or adapt to a loss of sensory feedback. Since our sensors may

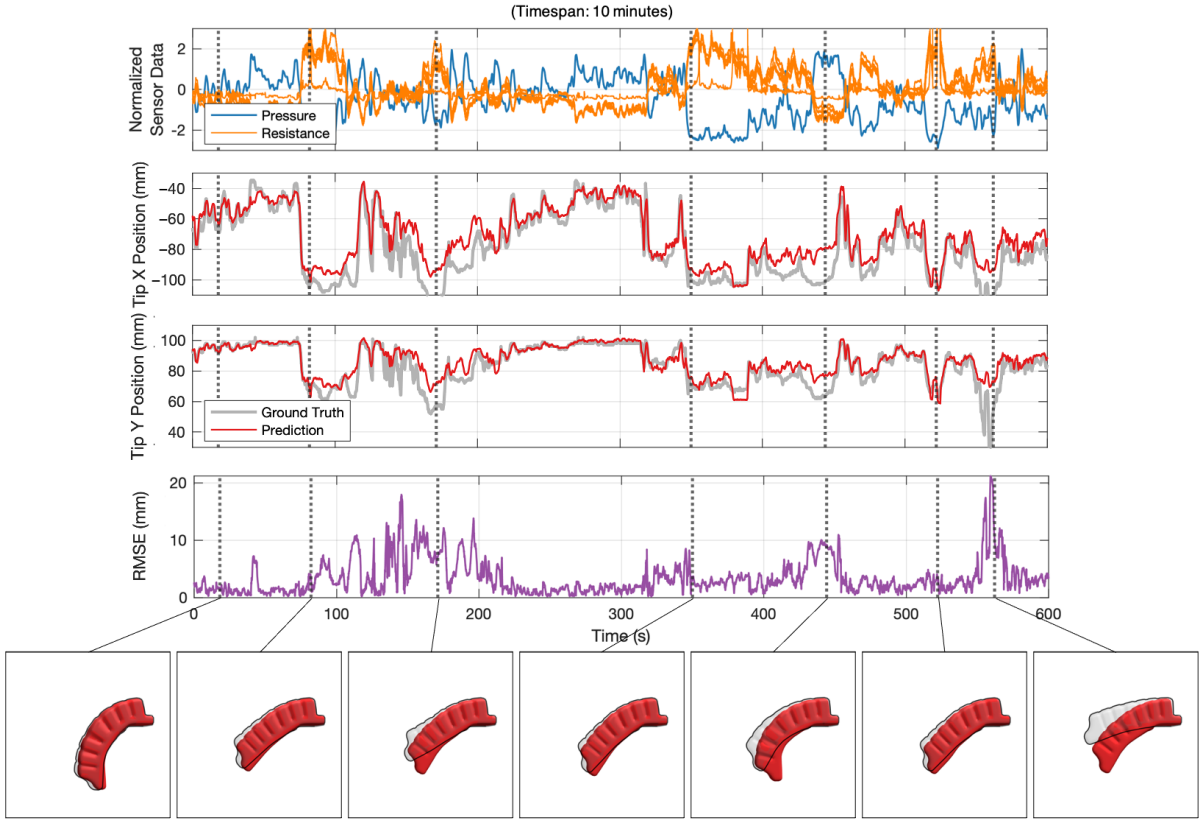

Figure 3.3: The entire 10 min of the test dataset is shown along with predictions from the BiLSTM model and snapshots of the resulting poses. The top row depicts the raw sensor data, showing both the single pressure sensor and the suite of resistances. The second and third rows show only the fingertip horizontal (X) and vertical (Y) positions. The fourth row is a plot of the root mean squared error across all the LED positions. The insets below depict the overall pose of the robot during seven moments in which the pose estimator was performing maximally or minimally well.

or may not have redundancy, we aimed to investigate the robustness of our BiLSTM model towards the loss of input data. Here, we attempted to artificially degrade the sensors by degrading the inputs from the test dataset and evaluated the resulting predictions from the previously trained model. In Figure 3.4, we present 4 min excerpts of the resulting outputs

of the trained model generated from input data that were gradually degraded by artificially saturating the inputs in turn. We saturated the sensors by setting the entire duration of the sensor's input to zero before z-score normalization. In the first case, we completely saturated the first two resistance-based sensors near the tip. The plot shows that, in general, the model still performed well. In the case with four sensors saturated, the model still predicted the movement well, but with an expected overall decrease in accuracy. However, with six sensors saturated, or with only the pressure readings intact, the model performed extremely poorly. It is notable that some dynamics were still captured using pressure alone, suggesting that the model leverages pressure readings to some extent. Conversely, with only the pressure readings removed and leaving all other sensors intact, the model performed considerably well. Arguably, this result shows that the input data may not be redundant after all, but this is difficult to determine due to the black-box nature of neural networks.

3.3.3 Network Architecture Choices

We also investigated other deep learning architectures that have been commonly suggested in the literature for time-series data and non-linear sensors, namely multi-layer perceptron (MLP), gated recurrent units (GRU) and long short-term memory (LSTM). To test the robustness of these models towards sensor degradation and compare them to BiLSTM, we first trained each respective model with clean input data from the training dataset; then, the resulting estimations from both clean and degraded input data from the test dataset were compared. We degraded the test data in several ways simultaneously: 1) the pressure data were saturated, 2) Gaussian noise was applied to one resistance-based sensor, and 3) the third resistance-based sensor from the tip was constantly saturated. The results shown in Figure 3.5 show that MLP fared worst out of all models tested, where Gaussian noise overwhelmed all outputs. GRU and LSTM-based models were more robust to noise but lost a considerable amount of accuracy, further solidifying our choice of using BiLSTM as our neural-network architecture.

3.4 Discussion

We have presented a proof-of-concept towards regarding the fabrication of a soft robotic actuator that does not require purposefully-built sensor structures. We show that our sensor–actuator hybrid can provide relatively accurate estimates of its pose despite input signals that originate from unstructured, noisy and time-lagged sensors. For many applications, it is reasonable to simply construct distinct, dedicated sensor structures alongside the actuator structures. A variety of research has demonstrated state estimation and closed-loop control using that approach (e.g., [142]). However, as the scale of the desired robot becomes smaller and as the number of desired sensors and actuators increases, simplicity and compactness become increasingly important. Combined sensor-actuators could serve as a strategy for dense and small sensors, with distinct engineering tradeoffs.

Sensors, especially in robotics, are commonly designed to have well-defined properties that are linear, repeatable and accurate. Most real-world sensors do not exhibit such behaviors under wide conditions, and those sensors that can be made to exhibit those characteristics require careful engineering and are often expensive. In biological systems, sensors are magnitudes worse than man-made sensors in most aspects, such as hysteresis and dependence on temperature, yet humans can perform intricate tasks in the noisiest of environments.

A key question for soft robotic sensing, then, is the degree to which the sensor interpretations should be based on an a priori model vs. empirical data and function approximation [10]. In this paper, we embrace the shortcomings of our sensors and pursue an empirical, data-driven approach. Our method is only one of a class of methods for using machine learning to make the most of complex soft robotic sensors [16, 92]. We use neural networks, which can be extremely powerful for time series regression and forward prediction tasks. However, as with all empirical techniques, they come at the cost of requiring a large number of examples along with relatively clean ground truth data. Their performance is not guaranteed to extrapolate data when starved of examples in particular scenarios—an example is shown in the last snapshot in Figure 3.3, where the particular pose was not present in the training

data. However, through the sensor degradation tests, we show robustness to other common types of faults.

In terms of practicality, our presented prototype still has several limitations. Firstly, using salt water as the actuation fluid increases the weight of the actuator greatly, especially when compared to its pneumatic equivalent. We hope to mitigate this in future prototypes by using the actuation fluid more effectively and putting emphasis on the design of the bellows themselves, such as the miniaturized bellows shown in [108]. The salt water solution used in our experiments is cheap and easy to prepare but is susceptible to evaporation due to the permeability of silicone rubber. This can be mitigated by using other solvents, as presented in other research [21]. Nevertheless, we believe that our methods are easily transferable to other solvents and actuator designs that use ionic liquids as the actuation medium.

Another limitation of the apparatus and experiment we show here is that it does not examine the effects of long-term cycling and time on the sensor performance. However, changes in sensor properties due to use do not appear to have a degrading effect during this experiment. Changes in the strain-resistance properties appear to be negligible or are perhaps accounted for by the machine learning model. This is demonstrated by the similar performance of the estimator at the beginning of the experiment and towards the end, as shown in Figure 3.3. Over time, the model predictions still hold with a minimal change in the overall pose output.

Hybrid sensor–actuators, combined with machine learning for signal interpretation, offer a path toward sensorized soft robots while adding minimal additional fabrication complexity. With this in mind, our immediate next step for future work will be to apply our technique for the closed-loop control of a soft sensor–actuator. Some challenges still need to be met, such as the time-lagged response to hydraulic actuation and the reduction in weight of the sensor–actuator. We believe that, at the least, our work serves as a proof-of-concept of our approach and will inform the selection of neural-network models for future studies related to soft robotic sensing and actuation.

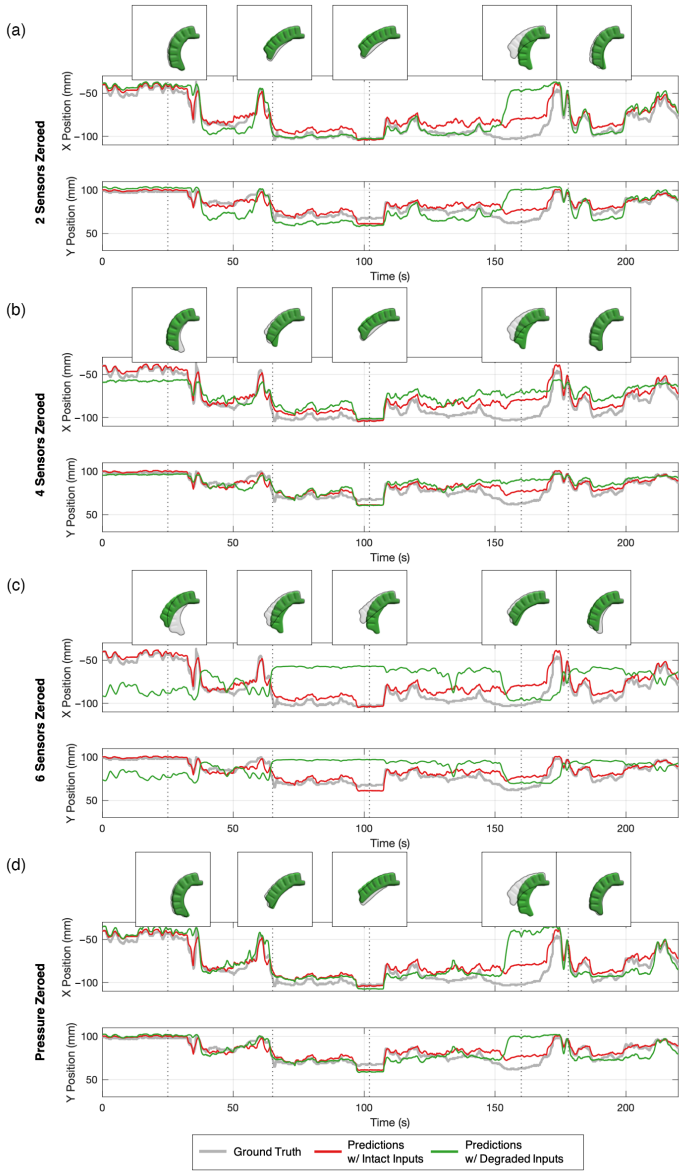

Figure 3.4: Plots of the tip position in X and Y in mm and snapshots of the pose of the sensor–actuator when estimated under various states of sensor degradation: (**a**) the voltage readings from the resistance-based sensors when the first two bellows from the tip of the sensor-actuator are saturated, (**b**) the first four sensors from the tip are saturated, (**c**) all six resistance-based sensors are saturated, leaving only the pressure reading intact, and (**d**) only the pressure sensor is saturated.

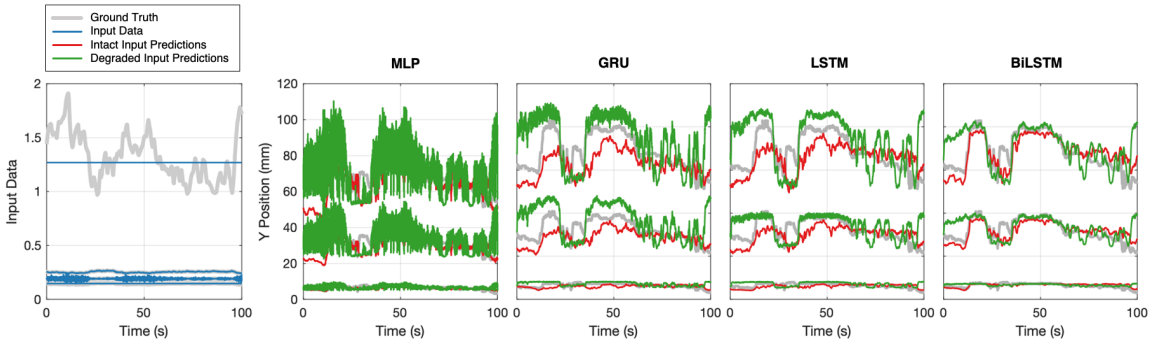

Figure 3.5: Results from comparing across models with the following conditions: 1) pressure data were constantly saturated, 2) Gaussian noise was multiplied to the data from sensor 2, and 3) sensor 3 was constantly saturated.

Chapter 4

CHASM: A SCREW BASED EXPRESSIVE COMPACT HAPTIC ACTUATOR

Pornthep Preechayasomboon[1], Ali Israr[2], Majed Samad[3]

1 Mechanical Engineering, University of Washington

2 ByteDance, formerly at Reality Labs, Meta

3 Google, formerly at Reality Labs, Meta

This chapter was originally published April 23, 2020 as a part of the Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems [106]

Abstract

We present a compact broadband linear actuator, Chasm, that renders expressive haptic feedback on wearable and handheld devices. Unlike typical motor-based haptic devices with integrated gearheads, Chasm utilizes a miniature leadscrew coupled to a motor shaft, thereby directly translating the high-speed rotation of the motor to the linear motion of a nut carriage without an additional transmission. Due to this simplicity, Chasm can render low-frequency skin-stretch and high-frequency vibrations, simultaneously and independently. We present the design of the actuator assembly and validate its electromechanical and perceptual performance. We then explore use cases and show design solutions for embedding Chasm in device prototypes. Finally, we report investigations with Chasm in two VR embodiments, i.e., in a headgear band to induce locomotion cues and in a handheld pointer to enhance dynamic manual interactions. Our explorations show wide use for Chasm in enhancing user interactions and experience in virtual and augmented settings.

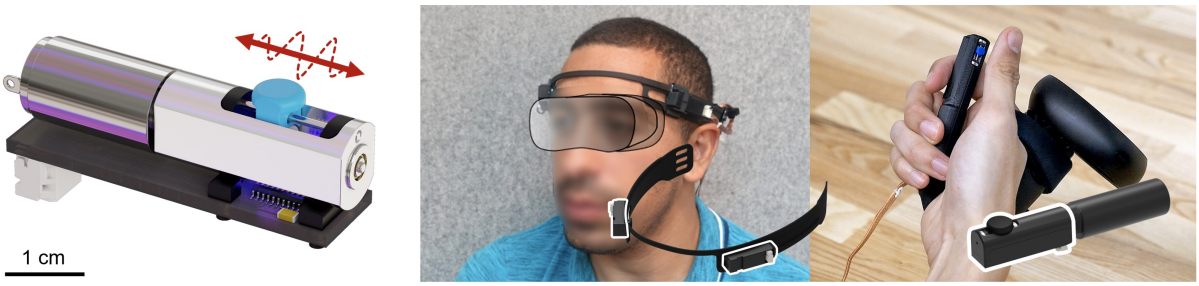

Figure 4.1: **(left)** A render of Chasm with a reference scale, the red arrow and wave represents how Chasm can simultaneously render both shear/normal forces and vibrations, **(middle)** A headgear prototype with two Chasm units on adjacent sides of the temple area on the head for enhancing locomotion in virtual reality (VR). A representative VR headset is shown. **(right)** A marker prototype with an embedded Chasm unit for enhancing interaction in augmented reality (AR) and virtual reality (VR), the hand is tracked with an attached Oculus Touch controller. The outlines show where Chasm is embedded in the prototypes.

4.1 Introduction

Human touch is critical for users' engagement with the surrounding environment and enables physical interactions with objects around them. Users reach out to pick objects, move them around and perform tasks while tuning their motor commands in response to dynamic interactions between the users' touch system and the objects' behavior. Despite this rich information flow between physical objects and the human touch system, current solutions to enable artificial touch feedback (or haptic feedback) are limited. Most commonly, vibrotactile actuators are incorporated in handheld devices and controllers to provide feedback of tactile contacts, impacts, button clicks, and surface texture [19] [53] [104]. The popularity of these actuators is due to their compact and lightweight design and low power requirements.

On the other end of the spectrum, low-frequency haptic devices that indent or stretch the skin can render percepts of dynamic behaviors of virtual objects comparable to the

inertia and stiffness of objects [37] [87] [112].The embedded actuators in these prototypes , however, have low bandwidth, are larger in size, have complex transmissions with non-linear system dynamics, and have relatively high-power requirements. Researchers have also accommodated both low-frequency dynamic effects and high-frequency vibrations in a single device by utilizing separate low- and high-frequency actuators to broaden the capacity and fidelity of haptic feedback [9] [17] [27] [67] [100].However, s imilar to the low-frequency actuators, these multi-dimensional actuators are bulky, and not practical for handheld and wearable applications.

In this paper, we introduce *Chasm*: a Compact Haptic Actuator with a Screw Mechanism. Chasm utilizes a common mechanical element, the leadscrew, which is found in many consumer grade devices. By coupling a leadscrew directly to a miniature high-speed DC motor, Chasm can render constant skin-shear as well as high-frequency vibrations both independently and simultaneously, therefore enabling high definition haptic feedback using a small actuator assembly. We present the design of Chasm and explore its use cases in a variety of wearable and handheld embodiments. We optimize the design of a leadscrew mechanism and embed it in (i) a head mounted system to induce navigational and locomotion cues and (ii) a handheld pointer to interact with broadband dynamic behaviors of virtual objects (Figure 4.1). In a series of engineering and user testing, we demonstrate high bandwidth and high fidelity of our design and its capacity to augment user interactions and experiences.

We present the following contributions:

1. We present the design and implementation of a compact screw-based one-degree-of-freedom haptic actuator capable of rendering broadband and high-fidelity haptic effects., e.g. low frequency skin-shear combined with high-frequency vibrations or independently.

2. We validate the actuator design with engineering and perceptual tests

3. We explore an interaction space and potential form factors using our device

4. We investigate two exemplary use cases and corresponding psychophysical evaluations to highlight their utility

4.2 Background

Mechanical stimulations on the body are felt through cutaneous (or tactile) and kinesthetic systems. *Cutaneous* receptors are embedded in the dermis and epidermis of the skin, and are activated by skin deformation due to indentation, stretch, texture, friction, and vibration. These receptors are sensitive to near DC to ∼500 Hz and peak at around 250 Hz, and can be activated by minuscule displacements and forces which saturate at a few millimeters of displacement and few Newtons of force [57]. *Kinesthetic* receptors are located in joints, muscles and tendons and detect joint movements and muscular activities. These receptors have small bandwidth (<20Hz) and operate in higher force and displacement ranges [83].

Common haptic actuators used in consumer devices are typically based on voice coil technology, such as the linear resonator actuator or LRA [19] and the eccentric rotating mass or ERM. Both actuators are small in size, have few components, and consume low power to create a noticeable haptic effect. They are used in game controllers, mobile devices, watches, and many other wearable and handheld devices. These actuators are designed to operate in the high-frequency vibration range (>100 Hz), where the sensitivity of human perception is high and low displacements are sufficient to create percepts. Driving these actuators at lower frequencies, however, may compromise with the overall efficiency such as size, weight, power consumption and heat dissipation. Piezoceramic based actuators are also available in compact formfactors, but they are also limited by high-frequency and low-displacement applications [104].

For low-frequency haptic feedback, recent research has utilized typical motors with custom transmission for skin stretch and skin indentation applications. Gear-trains or harmonic-drives first step-down the high motor speed and a custom mechanism converts the rotating shaft to linear motion. Prototypes have been built with pulley-cable [100] [105], belt [88], scotch-yoke [81], delta mechanisms, and gear-rocker [27]drives and then embedded in wear-

able and handheld devices. These devices can render high fidelity dynamics of virtual objects and produce sensations of movement, stiffness, inertia and squeeze [40] [100] [122]. Their use is however limited due to the relatively high number of components for transmission and therefore have higher cost, low reliability, audible noise between mechanical components, low output bandwidth, variability in performance in the operating range, and limited scalability to various formfactors.

Another compact haptic actuator is the Foldaway linkage by Salerno and colleagues [89], which uses an origami structure and micromotors to move a compact foldable platform against finger movements, hereby modulating the stiffness of virtual objects in contact up to 2 N. Recent research has also explored soft actuators, such as pneumatic and electroactive polymers for their compact and flexible housings, however, their use for wearable and handheld devices are limited due to oversized pressure sources (such as a compressor cylinder) or high driving voltages (beyond normal safety ranges). However, these actuators show potential for future use in wearable and handheld technologies [157].

A common rendering haptic primitive is *skin stretch*, which has been used to display friction [113]and inducing directional cues for driving guidance [103]. Skin stretch has also been used to display forces tangentially to finger pads, to simulate gravity [87] or other forces [94] [144].Tactors are placed in a pen type end effector to give force feedback and guidance cues [114], in a handheld device to create forces and torques [113],or mounted directly to the fingertip to render both contact force and weight [122]. In addition to the fingertip feedback, skin stretch can display information to the arms and other body locations [52] [67]. Recent research has also explored multidimensional haptic cues and sensory illusions in rich sensory environments and provided users with believable and expressive feedback [17] [27] [67]. Several researchers have also employed none to low-quality haptic feedback with altered visual interactions and showed enhanced perception of weight and other dynamic forces in virtual environments [73] [120]. Based on these studies, it is hypothesized that low profile skin deformations can substantially improve user interactions with objects in dynamic virtual environment, without the need of high force kinesthetic cues for a believable sensory

experience.

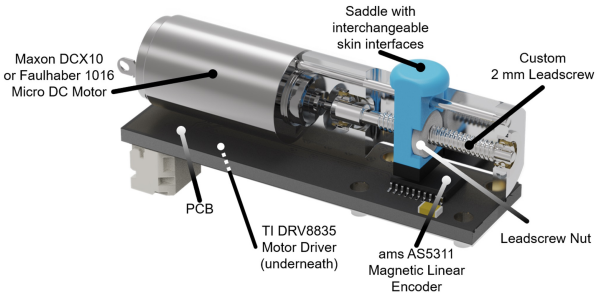

Figure 4.2: A cutaway section showing the mechanical and electrical components of Chasm

Recently, Sreetharan and colleagues investigated lead screw drives to directly convert the high rotational speed of a *stepper* motor to linear motion without the need of an intermediate transmission mechanism [132]. Their prototype was compact (8 mm 13 mm 27 mm; 5 grams) and had high output performance (max. force: ∼1 N, 5 mm stroke at ∼100 mm/s with 2 W of maximum power consumption). However, the performance of the actuator was substantially depreciated when a finger was loaded on the moving saddle, reducing both the stroke of the saddle and its bandwidth due to the open loop control scheme to command the motor [132].In this paper, we optimize the design of a screw directly-driven by a DC motor, specifically without the use of intermediate gearheads or step-down mechanisms, using a PI (proportional-integral) feedback control law to maintain high bandwidth and reduced finger loading effects, and explore its usability in a variety of wearable and handheld configurations.

4.3 Design

Chasm's core design element is a motor coupled directly to a leadscrew, that is, without any additional transmission or step-down mechanisms between the motor and the leadscrew. Chasm was also designed to have a minimal number of components: a DC motor without an integrated gearhead, a custom leadscrew and nut, two bearings, and a housing. Our design

process began by evaluating numerous leadscrew and motor manufacturers. Although some motor manufacturers offer motors with leadscrew shafts, these motors are often stepper motors which could not perform within our performance envelope [132]. Brushless direct current (BLDC) motors were also considered, but ultimately brushed direct current (DC) motors were chosen due to their simplicity to control and compact size. The leadscrew manufacturer, Ametek, was chosen mainly due to their capability in producing custom high-precision, low-friction leadscrews and leadscrew nuts. Through a process of evaluation of various combinations of leadscrews and motors, we finalized on pairing the 12 Volt Faulhaber 1016 [31] DC motor with a customized LSS series 2 mm leadscrew with a 1 mm lead [63], due to the satisfactory linear speed at nominal motor speed and excess in force output to accommodate for any frictional losses. With modularity in mind, the leadscrew nut has a simple flanged design with flats to accommodate 3D printed extensions. Figure 4.2 shows a sectional model of Chasm with the internal components visible. In order to obtain the high-fidelity haptic output performance in our design, we list the desired specifications in Table 4.1.

In order to track the position of the leadscrew, Chasm was designed to have a magnetic encoder placed directly underneath the leadscrew assembly. The Austria Microsystems (ams) AS5311 [5] was chosen due to its high bandwidth, high resolution and ease of integration. A magnetic strip with alternating poles every 1 mm serves as the tracker, which is bonded directly to the leadscrew nut. The Texas Instruments (TI) DRV8835 [51] was chosen to drive the motors due its compact size, low operating voltage and high current capability. Furthermore, with compactness and efficiency in mind, both the magnetic encoder and motor driver are assembled on to the same printed circuit board (PCB) that is mounted underneath the leadscrew and motor.

Lastly, a custom machined housing, two micro bearings and a 1 mm diameter shaft serve as the main structural elements. The bearings that are press-fit into the housing provide rigidity and constraints for normal and thrust loads for the leadscrew. The 1 mm diameter shaft constrains the nut from rotating while minimizing sliding friction. The overall size of

Chasm excluding the wiring terminal is approximately
11 mm 14 mm 45 mm.

Table 4.1: Specification of the desired and achieved Chasm actuator

Performance Metric	**Desired Value**	**Theoretical**	**Achieved**
Size and weight	<8 cm^3 and 25 grams	N/A	6.54 cm^3 and 15 g
Bandwidth	>100 Hz perceivable	N/A	170 Hz perceivable
Displacement	∼5 mm	5 mm	3.42 mm
Stroke rate	Linear up to 100 mm/s	110 mm/s	212 mm/s
Output Force	∼4 N	7.72 N	4.8 N
Noise	Unnoticeable audio	N/A	Minimal noise
Power Consumption	<2 W	2 W	Nominal: 0.6 to 1.44 W Peak: 2.7 W

4.3.1 Closed-loop Control Law

Chasm achieves closed loop control by reading the ams AS5311 magnetic encoder as input for the position of the leadscrew nut and sending correction commands to the TI DRV8835 motor driver. A PJRC Teensy 3.5 ARM Cortex-M4 microcontroller was used to maintain an interrupt-driven 1000 Hz proportional-integral (PI) controller on up to two Chasm units. The gains of the controller are selected to achieve a high gain crossover frequency for faster response time and a low overshoot, compensating for skin-loading disturbances and reducing structural and resonant harmonics in the DC to 100 Hz range.

4.3.2 Information Flow Framework

As Chasm is designed to be prototyped in various use cases, a robust software framework that can adapt to various scenarios is required. Firstly, in order to reduce guesswork in latency of

haptic rendering, we used the universal serial bus human interface device (USB HID) class for communicating between the rendering device and the Teensy microcontroller in "Raw HID" mode. USB HID has two major benefits: (i) it has plug and play capability for most devices on the market, including consumer VR devices such as the Oculus Quest, most Android smartphones and most modern PCs running Windows, Linux or macOS, which eliminates the need for custom drivers, (ii) USB HID requires the host device to reserve bandwidth for HID, which is only 64 bytes per millisecond, however, if a command packet is less than 64 bytes, the latency is ideally 1 millisecond.

Then, to increase scalability and aid development time, protocol buffers (protobuf) is used as a serialization layer, i.e. structured data is translated to protobuf binary data before sending or receiving the data between devices [44]. Consequently, typical packets being sent to the microcontroller, such as commands to move the end effector (leadscrew nut) to a specific position, are 6-10 bytes in size, which is well below the 64 bytes bandwidth for 1 ms latency. Another benefit is protobuf supports most popular programming languages, including C# and C++. We use the protobuf-unity [102] library for our Unity (C#) implementation and nanopb [4]for our microcontroller (C++) implementation.

Our implementation of the framework is as follows. On the firmware side, the microcontroller, while running the PI control loop, concurrently waits for a packet coming in from USB HID. The packet contains information such as the desired position for each Chasm unit, the speed in which to translate to that position, a command to play back stored sequences of positions or the sequences themselves (uploading waveforms). Once a packet is received and deserialized, the microcontroller immediately updates all state variables for the motors or performs the commanded actions. The microcontroller can also send data back to the host device using the same protocol for diagnostics. Figure 4.3 shows an overview of the framework.

Finally, we created operating system (OS) specific linked libraries (also known as plug-ins) for Windows, macOS, and Android. These plugs-ins can then be integrated into most applications that run on the system. In this paper, we exclusively use the Unity game engine

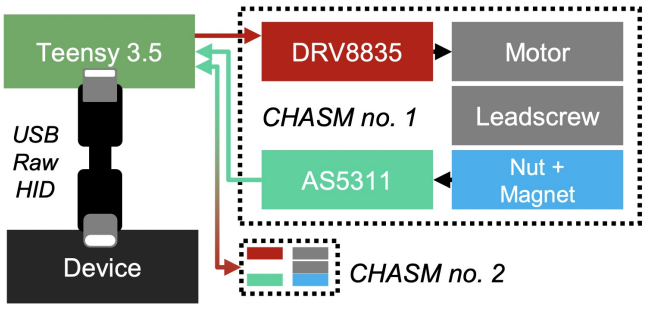

Figure 4.3: The overall architecture for Chasm's controller.

with our plug-ins because of its multi-platform and VR capabilities. Unity can then be used as a simulation engine to drive both the visuals and haptics, letting users and the actuator react to both modalities seamlessly.

4.4 Design Validation

With the proper control architecture in place, we can command the actuator to move the end effector with various displacement profiles under different loading conditions. To verify that Chasm, once integrated, can render various haptic effects from constant skin-stretch to high frequency vibrations, the following section presents results from experiments for validating the actuator's performance. We also summarize the comparison of Chasm's performance to other research prototypes in Table 4.2.

4.4.1 Temporal Response

The magnetic encoder was validated first to determine its usability as a sensor. We fitted a prototype of Chasm with a nut extension with a reflective marker, as shown in Figure 4.4a, and used a laser doppler vibrometer (Polytec PSV-500) to measure the actual displacement compared to the encoder readout. Figure 4.4b and Figure 4.4c show that the encoder has a linear response and can be trusted to provide accurate position tracking down to 100

Table 4.2: Comparison between Chasm's performance and similar research devices – unavailable data is represented with a dash

Device	**Vol. (cm3)**	**Wt. (g)**	**Workspace (mm)**	**Force (N)**	**Freq. (Hz)**	**Freq. -3dB (Hz)**	**Power (W)**
Chasm	6.54	15	3.4	4.8	170	20	0.6 – 2.7
Sreetharan [132]	1.87	5	6	1.5	15	4	<2
Tian [139]	121	100	18	3.23	-	-	-
Clemente [23]	52	60	6	8.6	30	5.5	-
hRing [94]	10.4	15	-	0.3	-	-	-
HapTip [36]	24.3	22	4×4	3.4	-	-	-
Schorr [122]	42	32	10×10×5	7.5	30	∼15	-
Foldaway [89]	188	130	35×35×30	2	140	20	-

microns. All subsequent experiments were performed using the magnetic encoder exclusively as positional tracking ground truth.

Unloaded Step Response. We ran a series of step response tests, in both the unloaded and loaded conditions. The actuator has an average rise time of 12.9 milliseconds and a steady-state error of 40 microns, as reported by the encoder and shown in Figure 4.4f. The top, no-load, speed of the actuator, as calculated from the rise time, is 217 mm/second.

Loaded Step Response. To put the actuator under nominal loading conditions, we first constructed mock prototypes with embedded capacitive force sensors to determine the usual normal loading. These mock prototypes reflect the use-cases in later sections and all loading conditions tested here are extreme loadings on the mock prototypes.

The loaded step response in Figure 4.4g shows that the actuator can still render displacement under a normal loading of 3.6 N with a constant displacement error of 150 microns.

Blocked Force Response. We also ran a blocked force response test, with the setup shown

in Figure 4.4d. The results, Figure 4.4e, show that the actuator can reach the steady state force of 4.8 N. The response characteristics can be further improved by tuning the derivative part of the PID control law. However, the design of a new and accurate control law is beyond the scope of this paper.

4.4.2 Frequency Response

To validate the extent of Chasm's bandwidth, we ran a series of frequency response experiments.

Chirp Response. Chasm was commanded a sinusoidal waveform that swept from 0.5 to 40 Hz at a constant amplitude of 1.71 mm (3.42 mm peak-to-peak) for a duration of 2 seconds. Figure 4.4h shows temporal response of Chasm's carriage displacement, and Figure 4.4i shows the frequency response with gain-crossover frequency around 20 Hz.

Sinusoidal Response. In a separate experiment, Chasm was commanded with sinusoidal frequencies from 1 Hz up to 100 Hz at amplitudes from 0.15 to 1.71 mm. The results shown in Figure 4.4j represent the mean attenuation across all amplitudes and loading conditions. The frequency response shows that Chasm has a significant attenuation above 20 Hz.

4.4.3 Human Perception Bandwidth

Although the results in the frequency response characterization show that the actuator's output is significantly attenuated at frequencies higher than 20 Hz, we ran a pilot study to show that rendered frequencies above 20 Hz can still be perceived, up to 170 Hz, as in Figure 4.5.

Seven participants, (7 males, aged 26 to 44) held a handheld prototype with their thumb resting on the actuator's end effector for measuring the finger's perceivability and a headgear prototype with the actuator touching the right side of the temple (as shown in Figure 4.1). Participants experienced two intervals of stimuli, either the first or second interval had the actuator active and participants would answer in which interval they felt the stimulus. The stimulus were displacements in a sine wave from 0.5 Hz to 170 Hz at a constant amplitude

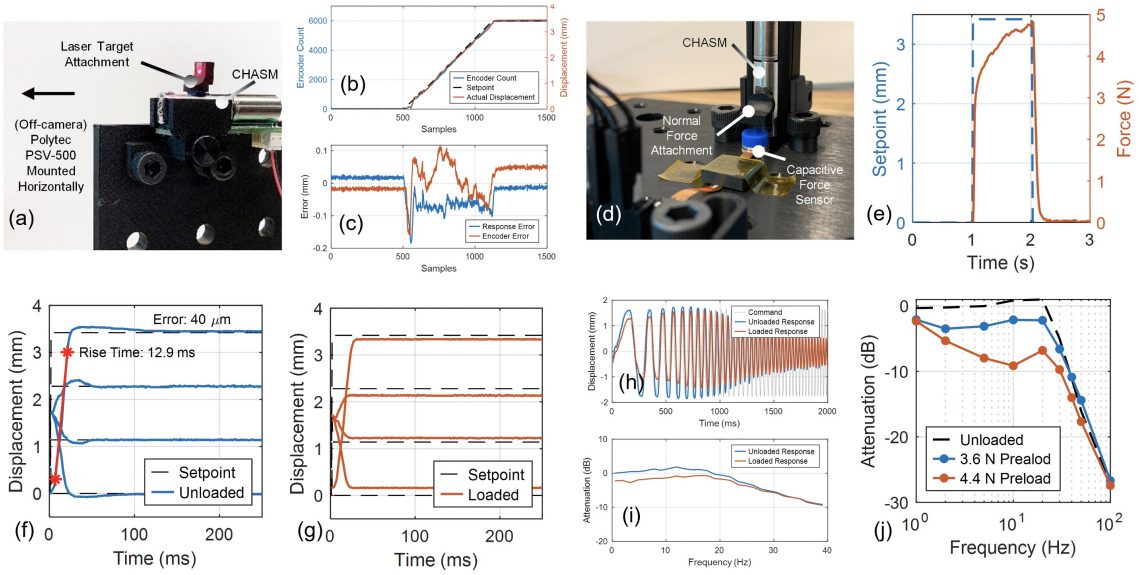

Figure 4.4: **(a)** The experimental setup for validating Chasm's encoder output **(b)** & **(c)** Plots showing the tracking error between Chasm's encoder and the laser vibrometer readings **(d)** The experimental setup for measuring Chasm's force output. **(e)** A plot showing the blocked force response given a step input of 3.4 mm **(f)** A plot showing the overall rise time of 12.9 ms, given an unloaded step input and the steady-state error of 40 microns **(g)** A plot showing the loaded step response with a 3.6 N preload **(h)** & **(i)** Plots showing the unloaded and loaded frequency response of a chirp signal from 0.5 to 40 Hz **(j)** A plot comparing the attenuation between the unloaded and loaded frequency response of 3.6 N and 4.4 N of preload at various frequencies up to 100 Hz

of 0.428 mm for 1 second. Participants also wore noise cancelling headphones with a tone played in the same frequency as the actuator at each interval to mask out any audible cues.

The results are shown in Figure 4.5 for the finger pad (red) and temple (blue). Even though the actuator's response is dampened significantly, it is still perceivable from 0.5 Hz to 170 Hz on the hand and from 0.5 to 72 Hz on the head. Beyond 128 Hz, participants

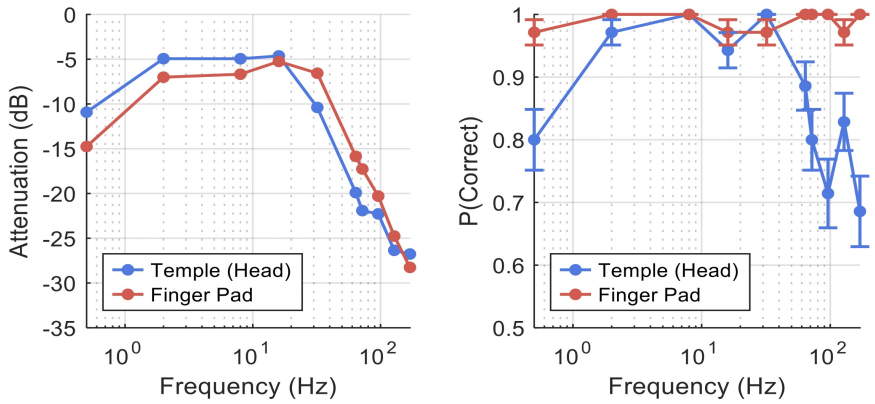

Figure 4.5: **(left)** A plot of the average attenuation when loaded on the head and finger at various frequencies. **(right)** The average responses of identifying the correct interval for each condition

reported that the whole device vibrated and was perceivable by the hand. This could be due to aliasing and the limited sample rate of our controller which is currently set to 1000 Hz.

4.4.4 Power Consumption

During the blocked force response experiment, we captured a peak current consumption of 225 mA at 12 Volts. Assuming a 60% output loss due to mechanical and electrical efficiencies, we can approximate the power density of Chasm to be 166 kW/m^3. However, during normal operation, such as rendering a 100 Hz sine wave at 0.25 mm amplitude or moving the end effector from resting to 2 mm, we observe only peak momentary current consumption from 50 to 120 mA. Furthermore, for certain conditions such as rendering a constant force, since Chasm's leadscrew is self-locking and non-backdrivable, we can simply drive Chasm to its intended position and cease to supply power since the position will be held regardless of power input.

4.5 Use Case Explorations

Chasm presents many advantages for haptic devices and is highly scalable. In this section, we highlight these advantages in a variety of embodiments and explore its design and interaction spaces. The benefits of Chasm are its broad bandwidth and flexible design configurations that enables the realization of high-fidelity haptic interactions embedded in a variety of wearable and handheld devices.

4.5.1 Wide Bandwidth and Linearity

Chasm has broad bandwidth compared to previously reported devices. We achieved this performance by using small and fewer components and optimize the actuator response by using a PI feedback control law that achieves high response time, operating range linearity, disturbance rejection, and near DC to >100 Hz perceivable frequency range. Many previous voice-coil solutions operate in broad frequency response from ∼30-500 Hz, which covers most of the vibrotactile range of human perception, however driving them at low-frequency movement profiles increase the size, power and heat generation. Due to low back drivability and the operating nature of the screw mechanism, Chasm's response is highly linear and consistent in its operating range; and can render both low-frequency (or low velocity) force-displacement profiles as well as high-frequency vibrotactile sensations, therefore extends into the broader dynamic range of human haptic perception.

4.5.2 Flexible Design Solutions

Compact, restricted and complex spaces

Chasm is housed in compact and small casing and can be directly integrated in hardware prototypes. For smaller profiles, a smaller size motor can be utilized, such as [132] achieved ∼1 N of force with a 5-gram design. In more restricted and complex spaces, screw mechanisms can be folded and/or beveled by using a pair of spur or bevel gears, as shown in Figure 4.6a. These gears have high efficiencies (up to 95%) due to small step-down speed and produce low

noise, and therefore compromises in the design extends the usability into many formfactors. Multiple Chasm units can be embedded in a device to distribute the load and interactions through independent contact locations with the device (Figure 4.6b), and screw linkages can also be a part of the overall structure of the device and provides resisting forces against user's loading.

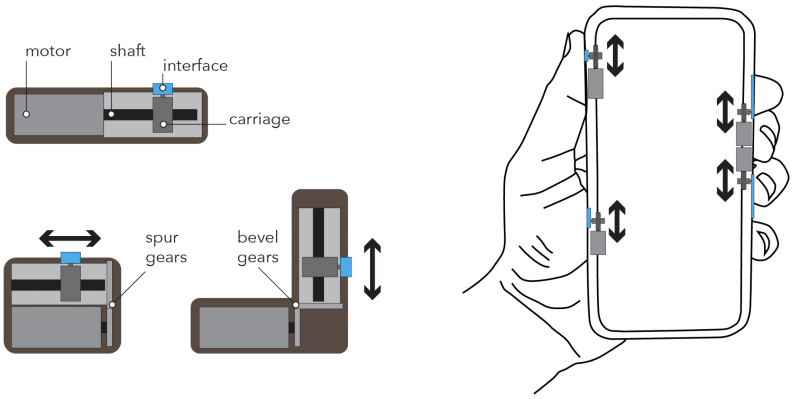

Figure 4.6: **(left)** Design variation of the screw mechanism. Direct drive lead screw **(top)** and folded and beveled designs (bottom) of the screw mechanism. **(right)** multiple screw mechanism are used in a cell phone case to distribute multi-finger interactions.

Multidimensional Configurations

In the design shown in the previous section, the rotational motor shape is translated to the linear carriage motion that is coupled to the skin interface. A typical application is sliding against the skin to shear or stretch the skin. A different coupling adapter can be used to translate the carriage motion to indent into the skin. Figure 4.7a shows a hand controller embedded with Chasm in shear mode to stretch the skin of the thumb, and Figure 4.7b shows Chasm in a normal force configuration to resist against the finger motion in a trigger-like action, where back drivability is emulated using an inline force sensor.

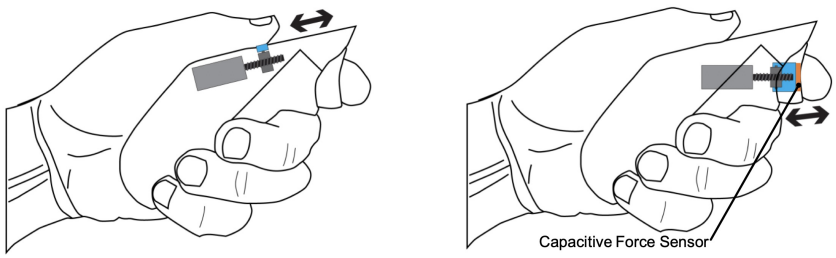

Figure 4.7: A hand controller embedded with Chasm in **(a)** shear mode and **(b)** normal mode, with an inline capacitive force sensor to help emulate back-drivability

4.5.3 Formfactors and User Experiences

Typical applications of Chasm are in *handheld* devices, where the linear motion of Chasm is interfaced with the skin of the fingers and palm and enhances dynamic interactions during manual tasks. It can be placed in a compact **game controller** or in the casing of a **cellphone** to render directional cues for navigations, surface texture during object explorations, stiffness and weight cues for object manipulation, dynamic shaking during collisions, and enhanced feedback with user interface widgets, such as in a button press or a slider motion. Another use case is a **stylus** shaped marker used as a writing device on flat surfaces and the embedded Chasm stretches the finger skin to render virtual textures and topology on the surface and alter mechanical features of virtual objects, as shown in Figure 4.8. Similar interactions can be done in mid-air with a pointing device, where the skin stretch of the finger enhances user interactions with pull-push like sensations and allow users to perform dynamic manual tasks, such as feeling compliance and weight of virtual objects – for example see [139].

Another utility of Chasm is in *wearable* devices, where the compact actuator stretches the skin to induce locomotion cues. In **shoes**, the actuator stretches the skin of the sole to indicate forward and backward motion while the user slides on virtual ice. A pair of Chasms

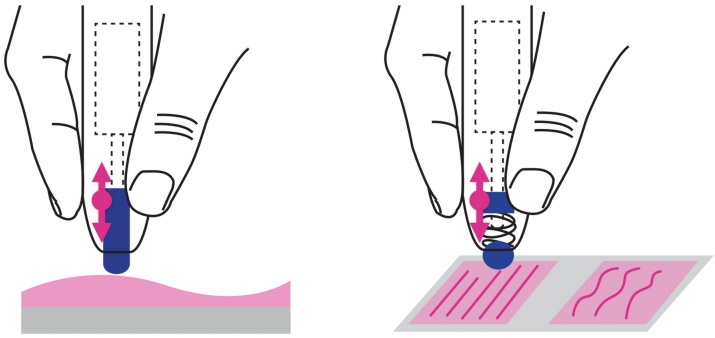

Figure 4.8: Other stylus-based use cases for a leadscrew-driven design. **(left)** Driving the height of a stylus tip. **(right)** Regulating the friction of a ball-point style tip via a spring.

in a **headband** stretches the skin of the temple on the forehead, and the coordinated motion of each actuator on two sides of the head induces illusion of translational and rotational motion of the head (Figure 4.9). The backward stretch of temple skin induces forward motion and the forward stretch induces backward motion. In addition to providing directional locomotion cues, stretch cues on the peripheral forehead could modulate motion sickness by inducing corresponding and reactive tactile cues to compensate for the high-speed optic flow through the visual sense.

4.6 Integration and Applications

In this section we discuss about the software architecture to quickly prototype experiences using Chasm and realize common wearable and handheld prototypes. We investigate the use of our actuator (i) in a head-mounted band to render haptic feedback at two sides of the temple to induce locomotion to users, and (ii) in a handheld pointing device to enhance interaction fidelity between the user and virtual objects, and explore use cases in these two settings with a series of user studies. We investigated if the inclusion of haptic cues improves user fidelity and performance bandwidth in rich virtual environments in a set of

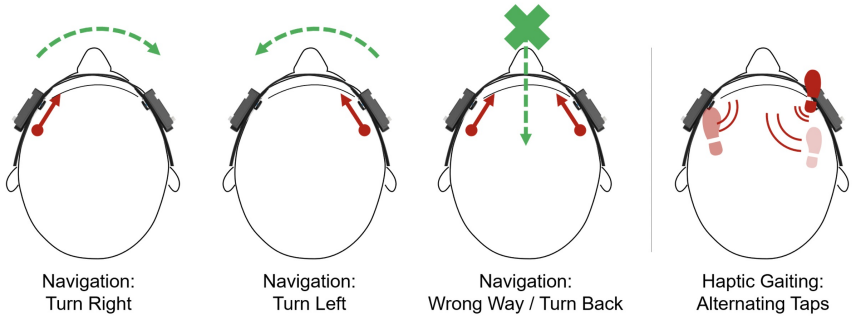

Figure 4.9: The four different haptic cues rendered by the head mounted device prototype

psychophysical studies presented in this section. All user studies are conducted after ethical and safety reviews by our institution.

4.6.1 Software Workflow for Interactive Gameplay

The Unity game engine, along with the USB HID plug-ins mentioned in the Information Flow Framework section, was used to create experiences for the Oculus Quest. On the development side, we created numerous tools to be used within the Unity Editor to author waveforms and commands that could be instantly tested on Chasm and later preloaded for rapid playback when needed. During real-time simulations, e.g. physics simulations, smaller commands such as "move the end effector to 0.5 mm at 0.5 mm/s" are sent to the controller, enabling responsive interaction scenarios that are directly coupled to the visuals presented. These commands and waveforms are encapsulated in our custom *HapticWaveform* protobuf message that contain data ready to be serialized and transmitted to Chasm's controller firmware. Once the experiences in Unity are ready to be tested or deployed, a compiled android application is sideloaded onto the Oculus Quest. Chasm's microcontroller, the Teensy 3.5, is then plugged directly into the USB Type C receptacle on the Oculus Quest to be used as a standalone device.

4.6.2 Head Mounted Gear to Induce Locomotion

We prototype the headgear by dismantling the head strap from a 3M H-700 series helmet and attaching it to a 3D printed nylon strip with rails on the edges and slots in the middle. We also printed a custom adapter and attached it to Chasm's housing such that the adapter slides on the headband on rails with the moving carriage extending beyond the slot, as shown in Figure 4.1. We tested several shapes, sizes, texture and material of the interface block, and selected a smooth circular disk (radius 6 mm) made from elastomeric polyurethane (EPU). This disk ensures that Chasm stretches the skin without making the user uncomfortable. We embed a pair of Chasm actuators on two adjacent sides of the headband and adjust the position of the interface to stretch the skin of the temple in the forward and backward direction as shown in Figure 4.9.

We ran two preliminary studies to investigate the impact of stretch cues on the forehead. Both studies were conducted in a VR environment, where participants wore the headband first to their comfortable levels and then donned an Oculus Quest headset over the band.

In the first study, participants were presented with an optic flow stimulus rendered as a star field moving towards the user and the speed of the optic flow was increased or decreased from a set value during a trial. Each participant's task was to indicate whether the star field motion accelerated or decelerated in the trial. In the visual only (V) condition, the optic flow was rendered at the reference speed of 20 m/s for 1 second followed by 3 seconds of acceleration or deceleration. In the visual-haptics (VH) condition, the visual cue was supplemented with stretch cues either both actuators move backwards (for acceleration) or forwards (for deceleration). Eleven values of change in optic flow speeds were tested (5 accelerated speeds, 5 decelerated speeds and one no change in speed) and the distance of the skin stretch was maintained constant for all VH trials.

Six participants (6 males, ages ranging from 26 to 44, median age: 27 years old) completed two test conditions (V and VH) composed of 88 trials, with 4 repetitions of each level, and the order of the conditions was randomized. Before the experiment, participants were instructed

and given a few trials to become familiar with the task and environment.

Figure 4.10 shows the "proportional of yes" response for participants feeling acceleration (y-axis) as a function of acceleration of optic flow in the two test conditions. The data is fitted with a psychometric curve using the Palamedes Toolbox [111]. The slope of the psychometric curve near the middle of the plot is steeper in VH condition than the V condition, suggesting better discriminability and multisensory integration in the presence of haptic cues, however, effects of haptic cues were not significant. Using the fitted psychometric curve, we evaluate the just-noticeable-difference (JND) and the point of subjective equality (PSE). Where the JND is 1.728 m/s^2 for the V condition is slightly improved to 1.359 m/s^2 for the VH condition. The PSE point for the V condition is 5.6 m/s^2 and the PSE for the VH condition is 6.114 m/s^2.

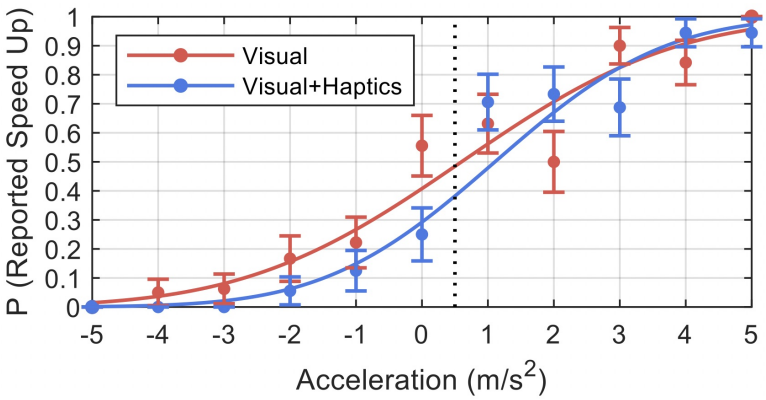

Figure 4.10: Psychometric plot comparing responses from the V and VH conditions for the acceleration study

In the second study, participants were placed in a VR maze of corridors and the task was to navigate through the maze to find the correct meeting room. An aerial map of the maze was provided to the user with the meeting room marked, however the orientation of the maze and location of the destination room were randomized in each trial, and location

of the participant was not indicated on the map. The map is provided in all conditions at the press of a button on the controller.

In no-haptics (No H) conditions, participants navigated through the maze using the map (Figure 4.11). In haptic-navigation (Nav H) conditions, navigation cues were provided to the participants through the headband. The right actuator stretching forward indicated the participant to turn left and the left actuator stretching forward would indicate a right turn. Both actuators stretching forwards means that the participant is headed the wrong way and should turn back, as shown in Figure 4.9. Participants were not told about the mapping of haptic feedback and were required to intuitively learn them and use them to complete the task. A third condition, haptic-gaiting (Gait H), was the same as the no-haptics condition except with alternating haptic cues on the two sides of the head corresponding to the gait of users' footsteps during walking, and the tempo of haptics cues was increased during running. The purpose of this condition was to evaluate if the peripheral haptic cues on the forehead have an effect on motion sickness in VR environments.

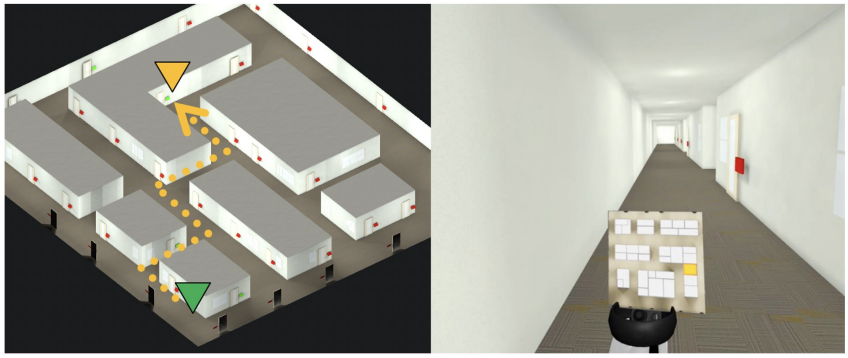

Figure 4.11: **(left)** An overview of the maze **(right)** The view from the VR headset while the participant views the map

Ten participants (9 males, ages ranging from 26 to 44, median age: 28 years old) com-

pleted the study. Each condition was presented once, and the order of conditions was randomized. Participants were allowed to look at the map as many times as needed and asked to reach the destination as quickly as possible. Participants started with a small corridor where they became familiar with the VR environment and the clock started once they crossed the corridor and entered the maze. Figure 4.11 shows a birds-eye-view of the map with the proposed directions and the participant's perspective.

Figure 4.12 shows the overall completion time and subjective evaluation of motion sickness induced due to the VR environment. The haptic-navigation condition yielded lowest completion times ($p<0.01$, paired Student's t-test), while there was no statistical difference between the other two conditions ($p>0.05$). The haptic-gaiting condition also induced lower subjective motion sickness. Participants reported that haptic effects were not annoying, but in fact pleasant and informative, and they were distracted to notice visual motion and were engaged with visual-haptic cues. Participants, however, felt that the head mount gear was tight especially after donning the VR headset on top.

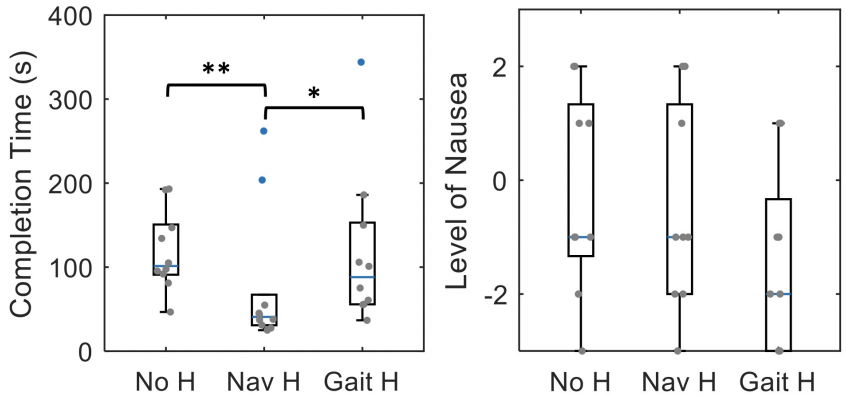

Figure 4.12: Box plots for completion time **(left)** and induced nausea level **(right)** while navigating in the VR maze.

4.6.3 Handheld Stylus for Enhanced Dynamic Interactions

Using a whiteboard marker as inspiration, we designed a housing for Chasm in a stylus formfactor. The front half of the housing houses Chasm's components while the other half is solid plastic sized to be similar to a whiteboard marker. Users hold the marker in the grip position shown in Figure 4.1 with their thumb resting on the end effector. The thumb would then experience shear force from the actuator creating the illusion of impact, stiffness, force, weight or texture. The body of the marker prototype was 3D printed from ABS plastic. To track the position of the marker to be rendered in VR, the user worn an Oculus Touch controller with a Velcro strap securing it around their palm, as shown in Figure 4.1.

Simulation Model

We used a "god object" model [164] to render haptic feedback on the thumb while the user interacted with objects using a virtual tool. The virtual tip of the marker, as seen by the user, cannot penetrate objects. When the actual position of the tip penetrates the virtual object, a force corresponding to the difference between the displacements of the virtual marker and the actual marker was rendered as skin stretch on the thumb. Figure 4.13 shows a simple implementation of a button press. When the tip of the real marker initiated the contact with the virtual button, the displacement of the button (d_B) and the displacement between the button and the tip (d_P) are both zero. As the tip pressed in the button, the resulting movement of the button in the visual scene is a function of the actual tip displacement. The ratio of the tip displacement (control, $C = d_B + d_P$) to the button displacement (display, $D = d_B$) induces the "pseudo-haptic" stiffness of the button, without any haptic cue [100, 120]. The movement of the virtual button was governed by its stiffness (k_B), and the resulting force feedback was rendered as $F_P = k_P d_P$. The pseudo-haptic ratio is:

$$C/D = \frac{d_P + d_B}{d_B} = 1 + \frac{k_B}{k_P}$$

The value of k_P was assigned to 2500 (N_v/m, where N_v is the Newton in game engine

units) to compensate with the scaling in the Unity Engine and to render k_B up to 10000 (N_v/m) by displacing to the extreme position of Chasm's moving carriage, i.e., a maximum skin-stretch of 3.42 mm. The carriage displacement (d_C) is linearly scaled to d_P and renders instantaneous forces (F_P) on the thumb.

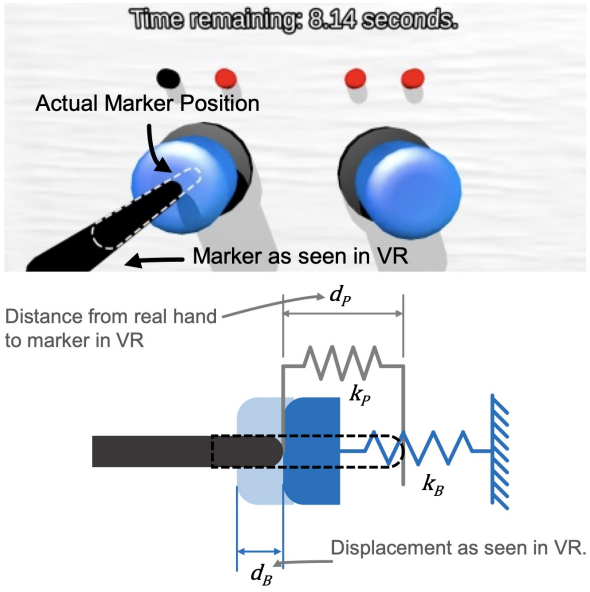

Figure 4.13: A simulation model for button pressing in VR

Psychophysical Study

In order to assess the usability of stretch cues on fingers for dynamic manual task, we ran a psychophysical study to evaluate the effectiveness of Chasm to render stiffness of a button press in the virtual environment. Thirteen participants (9 males, ages ranging from 25 to 44, median age: 28 years old, all righthanded) took part in the study. Participants were asked to sit on a chair, with the righthand controller strapped to the right hand holding the marker prototype and donned an Oculus Quest headset.

In each trial, participants were asked to use the marker to press into two buttons placed

side by side and identify the button with larger stiffness. A reference stiffness of 5000 N_v/m was randomly assigned to one button and the other button (target) was assigned the stiffness from one of 12 values with higher or lower stiffness between 1000-9000 N_v/m. Participants were told that they could feel each button twice and enter their response within the 10 second long trial. Each target stiffness was repeated four times with the total of 48 trials per test condition.

Three conditions were tested per participant: In the visual only (V) condition, pseudo-haptic feedback was presented with k_P set to 2500 N_v/m. In the haptic only (H) condition, participants felt skin stretch cues of reference and target stiffnesses through the marker displacement d_C, and k_P was set to infinity to obtain the C/D ratio of 1 for both button presses. In the visual-haptic (VH) conditions, both pseudo-haptic and congruent skin stretch cues were presented simultaneously. The levels and ranges of skin stretch and pseudo-haptic were determined in a pilot, where the extreme levels of stiffness were discerned $\sim$10% and $\sim$90% times from the reference stiffness. Participants completed all three conditions presented in a random order, therefore completing 144 trials which lasted roughly 30 minutes including breaks. Throughout the experiment, participants wore headphones playing pink noise.

The experimental responses were combined for all participants and proportion of "yes" responses that the button is stiffer were plotted against the target stiffness (Figure 4.14). Psychometric curves are fitted to the means and the error bars represent standard error of the means. A two way repeated measure ANOVA indicated significant effect of stiffness [$F(11,110)=10.9$; $p<0.001$] and test condition [$F(2,20)=6.6$; $p<0.01$] on the probability of yes response, while stiffness-condition interaction term was not significant ($p=0.55$). The slope of VH is much steeper than that of V, indicating better stiffness discriminability with the inclusion of haptic cues (Weber Fraction in V: 0.63 and VH: 0.297). In fact, improvement in sensitivity was entirely due to haptic cues (WF in H: 0.297) indicating effectiveness of skin-stretch in dynamical manual tasks.

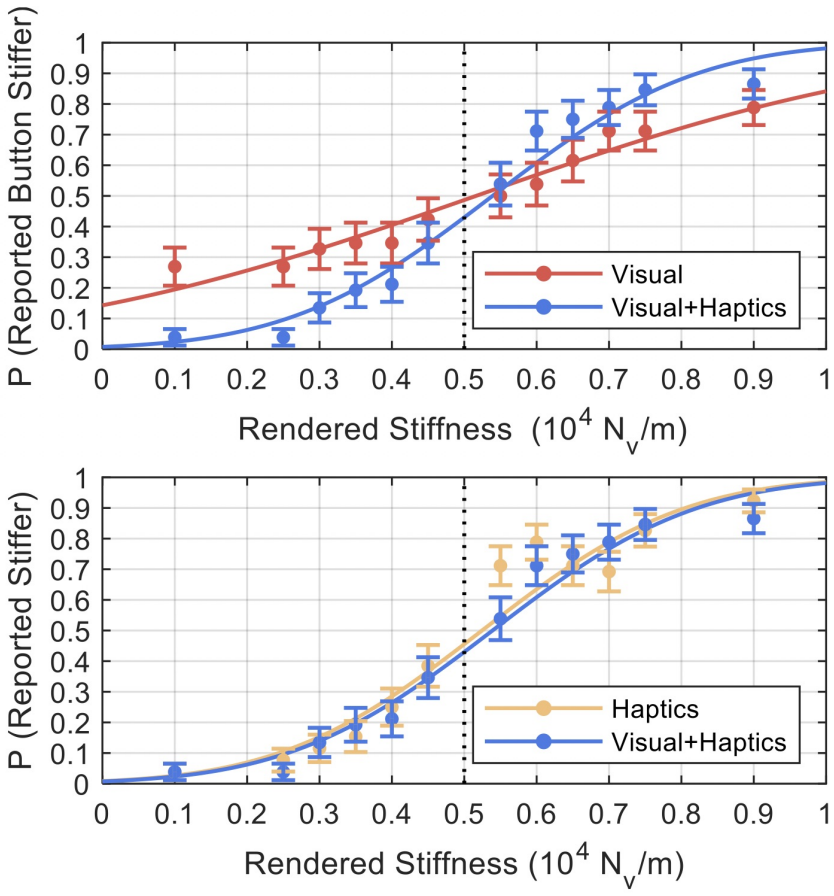

Figure 4.14: Psychometric plots comparing: **(top)** the V and VH conditions for the stiffness perception experiment and **(bottom)** the H and VH conditions

Interactive Scenarios

To evaluate the real-world use of Chasm's stylus prototype, we generated a set of scenarios in a VR experience presented as a series of "boards" the user can scroll through using two buttons (Figure 4.15). The user dons an Oculus Quest headset and holds the marker as shown in Figure 4.1. In all scenarios, the marker is represented as a black cylinder with a rounded tip. The marker cannot penetrate most objects presented and is tied to a virtual

spring to the actual marker location, creating a pseudo-haptic weight or force illusion upon contact similar to the technique used in the psychophysical study.

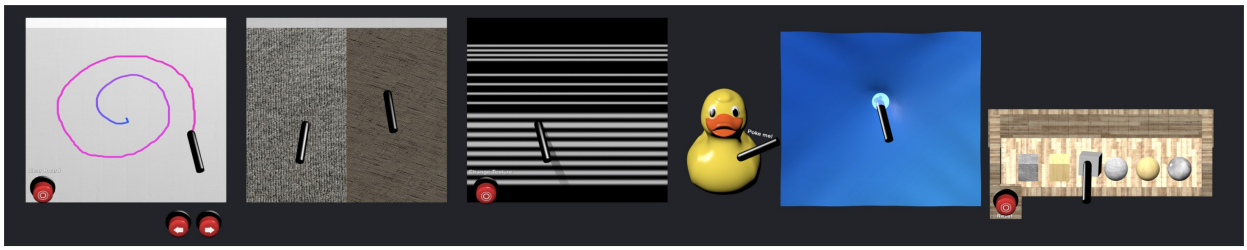

Figure 4.15: Demonstration boards for the for the marker prototype

Whiteboard. A virtual whiteboard is presented as a 1 × 1 meter square with a lightly textured surface. The user uses the virtual marker to draw on the surface of the whiteboard. Chasm renders a small "bump" upon contact with the board, simulating a constant pressure from touching the board. If the user starts to move the marker on the surface, a line is visually generated along with a vibration of 75 Hz. Once the user removes the marker from the board, another "bump" is rendered, simulating the release of pressure from the board.

Textures. Two different textures are presented side by side, one resembles a rough texture and the other resembles a smoother texture. Both textures have the same behavior as the whiteboard. The rough texture has a 25 Hz vibration cue while the smoother texture has a 40 Hz vibration cue.

Texture to Displacement Map. This experience serves a demonstration to encode visual textures (like colors or grey shades) to haptic cues. The marker's tip samples the texture in contact with it and commands Chasm's displacement corresponding to the sampled intensity in the red channel. Thus, the haptic texture can be authored using images, drawings and pictures. Alternatively, the user could see one texture being displayed and feel another invisible texture using a separate channel for enhanced haptic cues.

3D Model. A textured, large, 3D rubber ducky is rendered in the scene. The user pokes

the duck with the marker in order to create the presence of a physical object being in the scene. Chasm renders a displacement when the actual marker penetrates the duck, but the marker does not penetrate visually in order to create a force effect. The user moves across the model to feel curves and ridges of the 3D model.

Cloth Simulation. The user is presented with an orb that attaches to the marker when the marker is in proximity with the cloth. Once the orb attaches, Chasm renders a subtle haptic impact. As the cloth is stretched in, Chasm renders shear force accordingly. At the end of the cloth's stretch, the user penetrates through the cloth and a "snap" haptic effect is played. This experience demonstrates the use of arbitrary haptic waveforms that were authored as presets beforehand.

Cubes and Spheres. The user is presented with a shelf of 3 cubic blocks and 3 spherical balls of approximately 15 $\times$ 15 $\times$ 15 cm in size. Each object has a unique mass that reacts to the user prodding them with the marker according to Unity's physics engine. The user experiences both pseudo-haptic weight and the sensation of force from Chasm's displacement according to how far the virtual marker is to the actual marker.

4.7 Concluding Remarks

We present *Chasm*, a compact, lightweight and scalable screw based haptic actuator that can render dynamic haptic feedback in a variety of applications. Advantageous features of the actuator are its broadband range, linear operating features and low power consumption. Chasm renders multidimensional haptic feedback by directly converting the fast motor speed to the motion of the end effector, thereby simultaneously stimulating the skin with low-frequency skin deformations and high-frequency vibrations. In a series of studies and embodiments, we show that Chasm is used to stimulate forces, texture, vibrations in dynamical manual tasks, and induce navigational cues and illusory locomotion cues. In our future work, we will extend our preliminary findings of inducing forces and locomotion with Chasm and determine its capacity in other embodiments of wearable skin deformation devices to enhance user experience in high sensory environments.

Chapter 5

HAPLETS: FINGER-WORN WIRELESS AND LOW-ENCUMBRANCE VIBROTACTILE HAPTIC FEEDBACK FOR VIRTUAL AND AUGMENTED REALITY

Pornthep Preechayasomboon[1], Eric Rombokas[1]

1 Mechanical Engineering, University of Washington

This chapter was originally published September 20, 2021 in Frontiers in Virtual Reality, Haptics [109]

Abstract

We introduce Haplets, a wearable, low-encumbrance, finger-worn, wireless haptic device that provides vibrotactile feedback for hand tracking applications in virtual and augmented reality. Haplets are small enough to fit on the back of the fingers and fingernails while leaving the fingertips free for interacting with real-world objects. Through robust physically-simulated hands and low-latency wireless communication, Haplets can render haptic feedback in the form of impacts and textures, and supplements the experience with pseudo-haptic illusions. When used in conjunction with handheld tools, such as a pen, Haplets provide haptic feedback for otherwise passive tools in VR, such as for emulating friction and pressure-sensitivity. We present the design and engineering for the hardware for Haplets, as well as the software framework for haptic rendering. As an example use case, we present a user study in which Haplets are used to improve the line width accuracy of a pressure-sensitive pen in a virtual reality drawing task. We also demonstrate Haplets used during manipulation of objects and during a painting and sculpting scenario in virtual reality. Haplets, at the very least, can be used as a prototyping platform for haptic feedback in virtual reality.

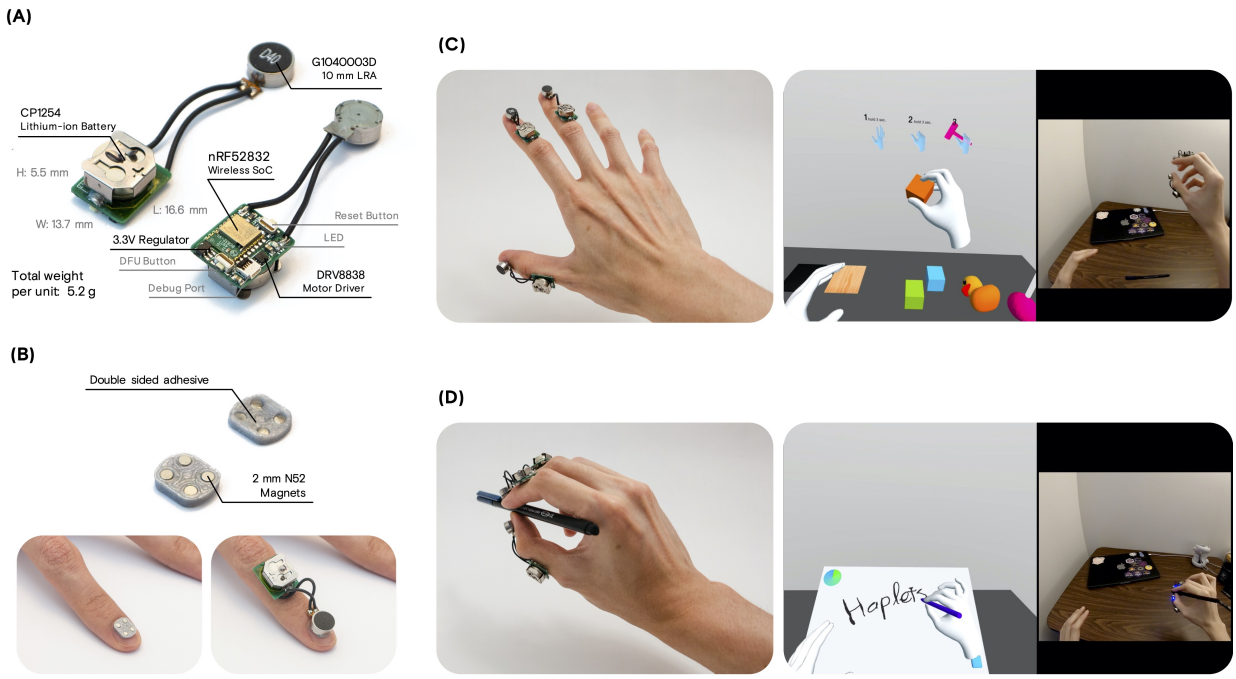

Figure 5.1: An overview of Haplets (A) Components comprising of a Haplet unit is shown, including the three key elements: the wireless SoC, the LRA, and the built-in battery. (B) Haplets are attached on the fingernails through fingernail-mounted magnets. The magnets are embedded in a plastic housing that is attached to the fingernail using double-sided adhesive. (C) The user wears Haplets on the thumb, the index finger and the middle finger. An example of a manipulation scenario as seen in VR compared to the real-world is shown. (D) When used with a tool (a pen, as shown), Haplets can be used to augment the virtual representation of the tool in VR by providing vibrotactile feedback in addition to the passive haptic feedback provided by the finger's grounding on the tool.

5.1 Introduction

Hands can be considered the most dexterous tool that a human naturally possesses, making them the most obvious input modality for virtual reality (VR) and augmented reality

(AR). In productivity tasks in AR and VR, natural hand tracking enables seamless context switching between the virtual and physical world i.e. not having to put down a controller first to interact with a physical keyboard. However, a major limitation is the lack of haptic feedback that leads to a poor experience in scenarios that require manual dexterity such as object manipulation, drawing, writing, and typing on a virtual keyboard [39]. Using natural hand input invokes the visuo-haptic neural representations of held objects when they are seen and felt in AR and VR [75]. Although haptic gloves seem promising for rendering a realistic sense of touch and textures, or provide kinesthetic impedance in the virtual or augmented space, wearing a glove that covers the fingers greatly reduces the tactile information from physical objects outside the augmented space. Thus having a solution that provides believable haptic feedback with the lowest encumbrance is desirable.

Numerous research devices have shown that there is value in providing rich haptic feedback to the fingertips during manipulation [56, 122, 46, 74], texture perception [12], stiffness perception [119], and normal and shear force perception [66, 106]. Although these devices may render high fidelity haptic feedback, they often come at the cost of being tethered to another device or have bulky electronics that impede the wearability of the device and ultimately hinder immersion of the VR experience. Additionally, once devices are placed on the fingertips, any interaction with objects outside the virtual space is rendered impossible unless the device is removed or put down. [134] has shown that wearable, wireless, low encumbrance haptic feedback on the fingertips is useful for AR scenarios with a prototype that leaves the fingertips free when haptic feedback is not required. Akin to the growing adoption of virtual reality, the device must be as frictionless to the user as possible — wearable haptic devices are no exception.

It has been shown that rendering haptic feedback away from the intended site does provide meaningful sensations that can be interpreted as proxies for the interactions at the hand [100], or for mid-air text entry [39]. [6] has shown that rendering vibrations on the fingernail can be used to augment passive touch-sensitive displays for creating convincing perception of edges and textures, others have extended this technique to include projector-

based augmented reality [116], and even used the fingernail as a haptic display itself [48]. We have shown that there is a perceptual tolerance for conflicting locations of visual and tactile touch, in which the two sensory modalities are fused into a single percept despite arising from different locations [11]. Furthermore, combining multiple modalities either in the form of augmenting otherwise passive haptic sensations [18], using pseudo-haptic illusions [2, 120], or a believable simulation [71, 12], can possibly mitigate the lack of congruence between the visual and tactile sensation. We therefore extend what [6] has proposed to immersive virtual reality by placing the haptic device on the fingernail and finger dorsum and compensating for the distant stimulation with believable visual and haptic rendering, which leaves the fingerpads still free to interact with real-world objects.

With the hands now free to hold and interact with physical objects, any passive object can become a tangible prop or tool. These held tools can provide passive haptic feedback while presenting familiar grounding and pose for the fingers. Gripmarks [162] has shown that everyday objects can be used as mixed reality input by using the hand's pose as an estimate to derive the object being held. In this paper, we further this concept by introducing Haplets: small, wireless and wearable haptic actuators. Each Haplet is a self-contained unit that consists of the bare minimum required to render vibrotactile stimulus wirelessly: a linear resonant actuator (LRA), a motor driver, a wireless system-on-a-chip (SoC), and a battery. Haplets are worn on the dorsal side of the finger and fingernail, and have a footprint small enough that the hands can still be tracked using computer vision methods. Combined with a believable simulation for rendering vibrotactile feedback in VR, Haplets can be used to augment the sensation of manipulation, textures and stiffness for bare hands while still maintaining the ability to pick up and handle everyday objects outside the virtual space. With a tool held in the hand, Haplets can render haptic effects to emulate the sensations when the tool interacts with the virtual environment. We use Haplets as an exploration platform towards building low-encumbrance, wearable haptic feedback devices for virtual and augmented reality.

The rest of this paper is organized as follows: first, in section 5.2, we describe the hardware

for each Haplet and engineering choices made for each component, including our low-latency wireless communication scheme. We then briefly cover the characterization efforts for the haptic actuator (the LRA). Then, we cover our software efforts in creating a physically-believable virtual environment that drives our haptic experiences, including physics-driven virtual hands and augmented physical tools. In section 5.3, we cover a small user study to highlight one use case of Haplets and explore the practicality of Haplets in a virtual reality scenario. In section 5.4, we demonstrate other use cases for Haplets in virtual or augmented reality environment such as manipulation, texture discrimination, and painting with tools. Finally, in section 5.5, we discuss our engineering efforts and the results of our user study, and provide insight for shortcomings and potential future work.

5.2 Materials and Methods

Haplets can be thought of as distributed wireless wearable haptic actuators. As mentioned previously, each Haplet consists of the bare minimum required to render haptic effects: an LRA, a motor driver, a wireless SoC, and a battery (Figure 5.1A). We minimized the footprint so that Haplets, aside from our target area of the finger, can be worn on other parts of the body such as the wrist, arms, or face, or integrated into other, larger systems. Haplets is designed to be able to drive other voice-coil based actuators such as voice coil motors (VCMs), eccentric rotating mass (ERM) actuators, and small brushed DC motors, as well.

5.2.1 Finger-mounted Hardware

The core electronic components of each Haplet, as shown in Figure 5.1A, are contained within one side of 13.7 mm by 16.6 mm PCB, while the other side of the PCB is a coin cell socket. We use a BC832 wireless module (Fanstel) that consists of a nRF52832 SoC (Nordic Semiconductor) and an integrated radio antenna. The motor driver for the LRA is a DRV8838 (Texas Instruments), which is chosen for its high frequency, non-audible, pulse width modulation limit (at 250 kHz) and versatile voltage input range (from 0 to 11 V). The PCB also consists of a J-Link programming and debug port, light emitting diode (LED),

and two tactile buttons for resetting the device and entering device firmware upgrade (DFU) mode. The DFU mode is used for programming Haplets over a Bluetooth connection. The coin cell we use is a Lithium-ion CP1254 (Varta), measuring 12 mm in diameter by 5.4 mm in height, which is chosen for its high current output (120 mA) and high power density. In our tests, Haplets can be used for up to 3 hours of typical usage and the batteries can be quickly replaced. The total weight of one Haplet unit, including the LRA, is 5.2 grams.

We imagine Haplets as a wearable device, therefore Haplets must be able to be donned and doffed with minimal effort. To achieve this, we use 3D printed nail covers with embedded magnets, as shown in Figure 5.1B, to attach the LRA to the fingernail. The nail covers are small, lightweight, and can be attached to the fingernail using double-sided adhesive. Each cover has a concave curvature that corresponds to each fingernail. The Haplets' PCB is attached to the dorsal side of the middle phalanx using silicone-based, repositionable double-sided adhesive tape. In our user studies and demonstrations, we place the Haplets on the thumb, index finger and middle finger of the right hand, as shown in Figure 5.1C.

5.2.2 Low-latency Wireless Communication

Since we target the fingers, we desire to reduce the latency from visual perception to tactile perception as much as possible, especially when considering the mechanical time constant of the LRA[1]. Although Bluetooth Low Energy (BLE) is commonplace and readily available in most systems with a wireless interface, the overall latency can vary from device to device. We therefore opted to use Enhanced Shock Burst (ESB)[2], a proprietary radio protocol developed by Nordic Semiconductor, for our devices instead. ESB enables up to eight primary transmitters[3] to communicate with a primary receiver. In our implementation, each Haplet is a primary transmitter that sends a small packet at a fixed interval to a host microcontroller,

[1] `https://www.vibration-motor.com/wp-content/uploads/2019/05/G1040003D.pdf`

[2] `https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/ug_esb.html`

[3] It is worth noting that although this suggests that a maximum of eight Haplets can be communicating with one host microcontroller at once, there exists techniques such as radio time-slot synchronization similar to those used in Bluetooth that can increase the number of concurrent transmitters to 20.

a primary receiver, which is another SoC that is connected to a VR headset, smartphone or PC (Figure 5.2A). Commands for each Haplet are sent in return along with the acknowledge (ACK) packet from the host device to the Haplets. If each Haplet transmits at an interval of 4 milliseconds, then ideally the maximum latency will be slightly over 4 milliseconds when accounting for radio transmission times for the ACK packet.

Since Haplets transmit at a high frequency (every 4 ms or 250 Hz), there is a high chance of collisions between multiple units. We mitigate this by employing a simple time-slot synchronization scheme between the Haplets and the host microcontroller where each Haplet must transmit in its own predefined 500 microsecond timeslot. The host microcontroller keeps a 250 Hz clock and a microsecond counter that resets every tick of the 250 Hz clock. The counter value from the host is transmitted along with the command packets and each Haplet then uses the counter value to adjust its next transmitting interval to correct itself. For instance, if a Haplet receives a counter value of 750 microseconds and its timeslot is at 1000 microseconds, it will delay its next round of transmission by 250 microseconds or 4250 microseconds in total, after the correction, it will transmit at the usual 4000 microsecond interval until another correction is needed.

One drawback of our implementation, as we use a proprietary radio protocol, is we cannot use the built-in Bluetooth capabilities of host devices (i.e. VR headsets or PC) to communicate with Haplets. Therefore, we use a nRF52840 SoC (Nordic Semiconductor) as a host microcontroller that communicates with the host device through a wired USB connection. In order to minimize the end-to-end latency, we use the Human Interface Device (HID) class for our USB connection. The benefits are two-fold: 1. HID has a typical latency of 1 millisecond and 2. HID is compatible out-of-the-box with most modern hardware including both Windows and Unix-based PCs, standalone VR headsets such as the Oculus Quest, and most Android-based devices [106].

We briefly tested the communication latency of our system by running a test program that sends command packets over HID to our host microcontroller to three Haplets at 90 Hz — this frequency is chosen to simulate the typical framerate for VR applications. Two

digital output pins, one from a Haplet and one from the host microcontroller, were connected to a logic analyzer. The Haplet's output pin toggles when a packet is received and the host microcontroller's output pin toggles when a HID packet is received. Therefore, latency here is defined by the interval of the time a command is received from the host device (PC) to the time the Haplet receives the command. We found that with three Haplets receiving commands simultaneously, the median latency is 1.50 ms over 10 seconds, with a maximum latency of 3.60 ms during our testing window. A plot of the latency over the time period is shown in Figure 5.2B along with an excerpt of captured packet times with the timeslot correction in use in Figure 5.2C. It should be noted that the test was done in ideal conditions where no packets were lost and the Haplets are in close proximity with the host controller.

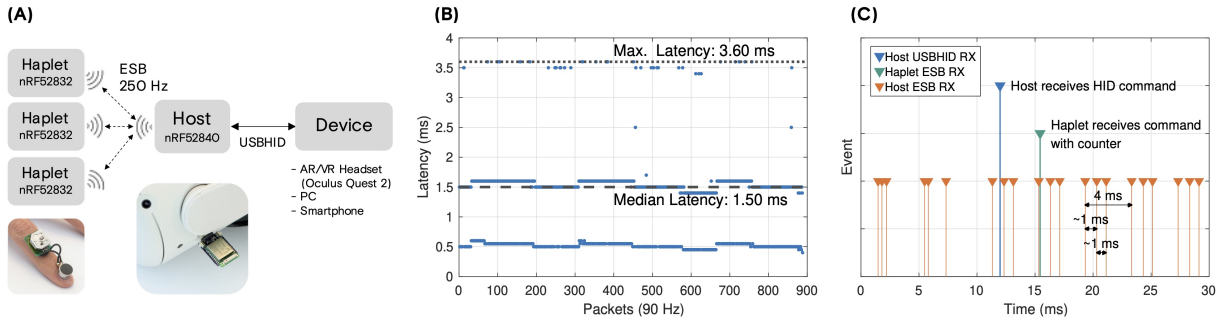

Figure 5.2: Haplets' low-latency wireless communication architecture. (A) Each Haplet communicates using the Enhanced ShockBurst (ESB) protocol with a host microcontroller that receives command from a host device using USB HID. Commands are updated at the rate of 250 Hz. (B) Latency, as defined by the time the host microcontroller receives a command from the host device to the time a Haplet receives the command, is shown over a 10 second interval. The maximum latency and median latency is 3.60 milliseconds and 1.50 milliseconds, respectively. (C) Events received from our logic analyzer showing our timeslot algorithm momentarily adjusting the period for sending packets over ESB to prevent radio collisions between Haplets.

5.2.3 Vibration Amplitude Compensation

Haplet's LRA is an off-the-shelf G1040003D 10 mm LRA module (Jinlong Machinery & Electronics, Inc.). The module has a resonant frequency at 170 Hz and is designed to be used at that frequency, however, since the LRA is placed in such close proximity to the skin, we observed that frequencies as low as 50 Hz at high amplitudes were just as salient as those closer to the resonant frequency at lower amplitudes. Lower frequencies are important for rendering rough textures, pressure, and softness [71, 18] and a wide range of frequency is required for rendering realistic textures [33]. Thus, we performed simple characterization in order to compensate for the output of the LRA at frequencies outside the resonant frequency range, from 50 Hz to 250 Hz. Figure 5.3C shows the output response of the LRA as supplied by the manufacturer when compared to our own characterization using the characterization jig in Figure 5.3A (as suggested by the Haptics Industry Forum[4]), and when characterized on the fingertips (Figure 5.3B). The acceleration output was recorded using a micro-electromechanical-based inertial measurement unit (MEMs-based IMU) (ICM42688, TDK) on a 6.4 mm by 10.2 mm, 0.8 mm thick PCB connected to a specialized Haplet via an I^2C connection through a flat flex ribbon cable (FFC). The specialized Haplet streams readings from the IMU at 1000 Hz to the host device for recording on a PC. We found that when using the compensation profile derived from our characterization jig (Figure 5.3A) on the fingernail, the output at higher frequencies were severely overcompensated for and provided uncomfortable levels of vibration. However, when characterization was performed at the target site (the fingertips), with the IMU attached on the fingerpad, the resulting output amplitudes after compensation were subjectively pleasant and more consistent to our expectations.

Characterization was performed by rendering sine wave vibrations at frequencies ranging from 50 Hz to 250 Hz in 10 Hz increments and at amplitudes ranging from 0.04 V to 1.9 V (peak-to-peak) in 0.2 V increments with a duration of 0.5 seconds. Acceleration data was

[4]High Definition Inertial Vibration Actuator Performance Specification `https://github.com/HapticsIF/HDActuatorSpec`

sampled and collected at 1000 Hz during the vibration interval. A total of 5 repetitions of the frequency and amplitude sweeps were performed. The resulting amplitude is the mean of the maximum measured amplitude of each repetition for each combination of frequency and amplitude, as presented in Figure 5.3D. The output compensation is then calculated first by fitting a linear model for each frequency's response, $amplitude = x_f V$. Then, the inverse of the model is used with the input being the desired acceleration amplitude, in m/s^2, and the output being the voltage for achieving that acceleration. Frequencies outside the characterized models are linearly interpolated. The results from our compensation on a subset of the characterized frequencies, along with frequencies outside of the characterization intervals, is shown in Figure 5.3E. We use the same compensation profile for every instance of haptic rendering throughout this paper.

5.2.4 Virtual Environment

Our haptic hardware is only one part of our system. Robust software that can create compelling visuals and audio as well as believable and responsive haptic effects is equally important. In this paper, we build up a software framework using the Unity game engine to create our virtual environments as described in the following sections. Our entire system is run locally on the Oculus Quest 2 and is completely standalone: requiring only the headset, our USB-connected host microcontroller and the Haplets themselves.

Haptic Rendering

Haplets are commanded to render sine wave vibrations using packets that describe the sine wave frequency, amplitude and duration. Each vibration becomes a building block for haptic effects and are designed to be either used as a single event or chained together for complex effects. For example, a small "click" that resembles a click on a trackpad can be commanded as a 10 millisecond, 170 Hz vibration with an amplitude of 0.2 m/s^2. To render textures, short pulses of varying frequencies and amplitudes are chained together in rapid succession. Due to the low-latency, haptic effects can be dynamic and responsive to the environment.

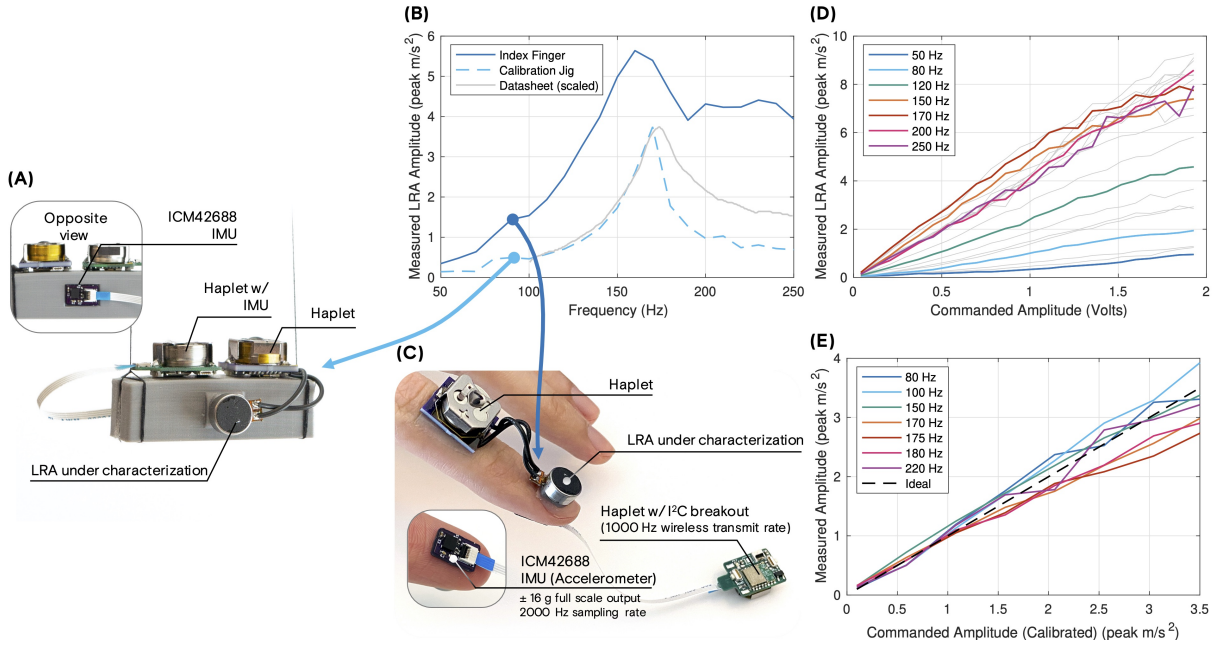

Figure 5.3: Haplets under characterization are shown and their resulting plots. (A) The characterization jig used for characterizing a Haplet's LRA. The jig is hung using two threads from a solid foundation. A special Haplet with an IMU is used to record the accelerations resulting from the LRA's inputs. (B) The characterization results from the jig closely resembles the characterization derived from the LRA's datasheet, however, the characterization results when the LRA is placed on a fingernail is substantially different. (C) The same devices used in the characterization jig are placed on the fingernail and fingerpads to perform LRA characterization at the fingertips. (D) Results from characterization at the fingertip at various frequencies. (E) After compensating for reduced amplitude outputs using the model derived from characterization, commanded amplitudes, now in m/s^2, closely match the output amplitude.

Examples of interactions that highlight the responsiveness of such a low-latency system are presented in the following sections.

Physics-Driven Hands

The user's hands in our environment are physically simulated using NVIDIA PhysX Articulations system for robotics [5]. Articulations are abstracted as ArticulationBodies in the Unity game engine. This system enables robust hand-object manipulations and believable response towards other rigid bodies in the scene, such as pushing, prodding, and throwing. The fingers are a series of linkages connected using either 1, 2, or 3 degree of freedom revolute joints with joint limits similar to that of a human hand [24]. The wrist is connected to the tracked position of actual wrist using a 3 degree of freedom prismatic joint. As a result, pseudo-haptics [79] is readily available as part of the system, meaning that users must extend their limbs further than what is seen in response to a larger force being applied to the virtual hands. This is also known as the pseudo-haptic weight illusion [120] or the god object model [164]. For higher fidelity, we set our simulation time step to 5 ms and use the high frequency hand tracking (60 Hz) mode on the Oculus Quest 2. A demonstration of the system is described in Figure 5.4.

We take advantage of the robust physics simulation and low-latency communication to render haptic effects. A collision event that occurs between a finger and an object is rendered as a short 10 ms burst of vibration with an amplitude scaled to the amount of impulse force. Each object also has unique haptic properties: the frequency of vibration during impact with fingers and the frequency of vibration during fingers sliding across the object. For instance, a wooden surface with high friction would have a sliding frequency of 170 Hz and a rubber-like surface with lower friction would have a sliding frequency of 200 Hz. Additionally, as both our objects and fingers have friction, when a finger glides across a surface, the stick-slip phenomenon can be observed both visually and through haptic feedback (Figure 5.5A & B).

---

[5] `https://gameworksdocs.nvidia.com/PhysX/4.0/documentation/PhysXGuide/Manual/Articulations.html`

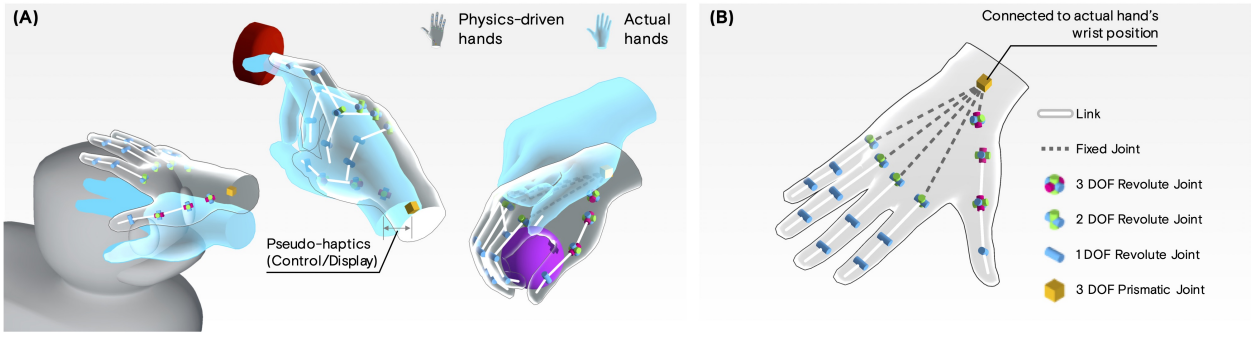

Figure 5.4: Physics-driven hands. (A) Physics-driven hands shown in the following scenarios from left to right: (1) When pressing against a stationary object the physics-driven fingers conform along the object's curvature while respecting joint limits. (2) When pressing on a button with a spring-like stiffness, the fingers do not buckle under the constraints. The whole hand is also offset according to the force resisting the hand, resulting in a pseudo-haptic illusion. (3) When grasping and lifting objects, the fingers respect the geometry of the object and conform along the shape of the object. Gravity acting on the object and inertia also dictates the pseudo-haptic illusion. (B) A diagram showing the articulated bodies and their respective joints. (The hand's base skeleton is identical to the OVRHand skeleton provided with the Oculus Integration SDK)

Tools

With the fingerpads free to grasp and hold actual objects, we augment the presence of handheld tools using visual and haptic feedback.

First, we detect the tool being held in the hand using a technique similar to template matching, as presented in GripMarks [162]. Each tool has a unique pose of the hand, such as the pose when holding a pen or the pose when holding a spray bottle (Figure 5.6), which is stored a set of joint angles for every joint of the hand. Our algorithm then compares each tool's predefined pose to the current user's pose using the Pearson correlation coefficient in

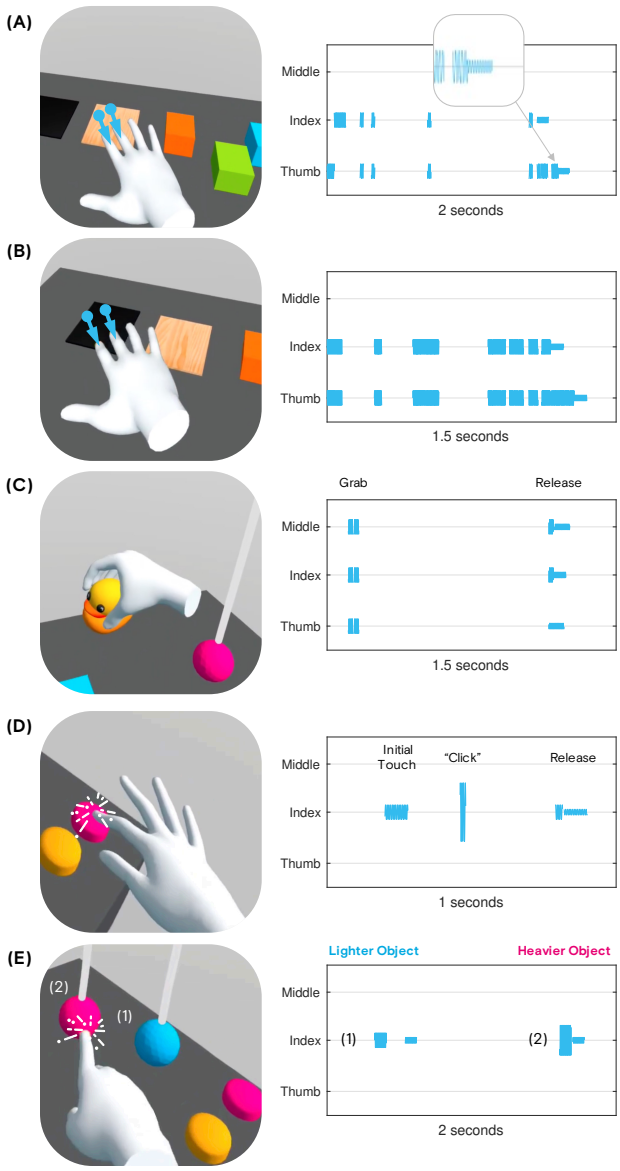

Figure 5.5: Haptic rendering output, shown as waveforms, during a time window of interaction in the following scenarios: (A) dragging fingers across wood, (B) dragging fingers across smooth plastic, (C) picking up and letting go of a plastic toy, (D) pressing a button, and (E) prodding two similar spheres with different masses.

a sliding 120 frame window. If 80% of frames in the window contains a pose with over a coefficient over 0.9, then it is deemed that the tool is being held in the user's hand. To "release" a tool, the user would simply open their hand fully for 1 second (Figure 5.6). Any tool can now be altered in shape and experience both visually and through haptics through the headset. For instance, the user can physically hold a pen but in a pose akin to holding a hammer, and in their VR environment, they would see and feel as if they are holding a hammer. Additionally, since our algorithm relies only on the hand's pose, an actual tool does not have to be physically held by the user's hand, we also explore this in our user study in the following sections.

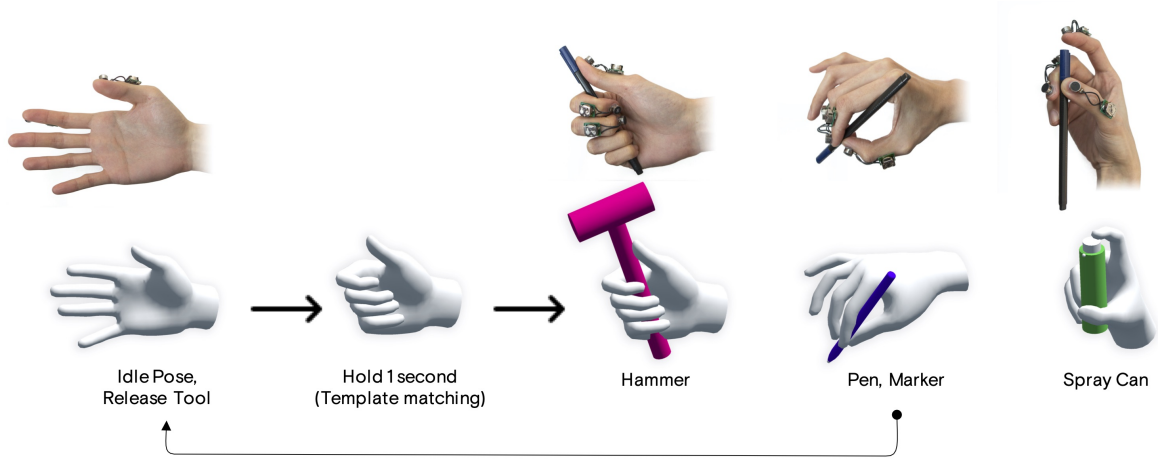

Figure 5.6: An illustration of the tool activation system. The user starts with an open hand and holds the desired pose for each tool for 1 second. The user can also hold a physical proxy of the tool in the hand. After 1 second, a tool is visually rendered in the virtual hands and the haptic rendering system augments any interaction of the tool with environment. To release a tool, the user fully opens their hand for 1 second. Switching between tools requires the user to release the current tool first.

When a tool is detected, the tool is visually rendered attached to the hand. In our physics

simulation, the tool's rigid body is attached to the wrist's ArticulationBody and thus the tool can respond dynamically to the environment as if the tool and the hands are a single object, maintaining the same pseudo-haptic capabilities as presented in the previous sections. The physically held tools provide passive haptic feedback in the form of pressure and familiar grounding while Haplets can be used to render vibrotactile feedback to augment the presence of the held tool in response to the virtual environment. Three examples of haptic rendering schemes for tools are presented in Figure 5.7.

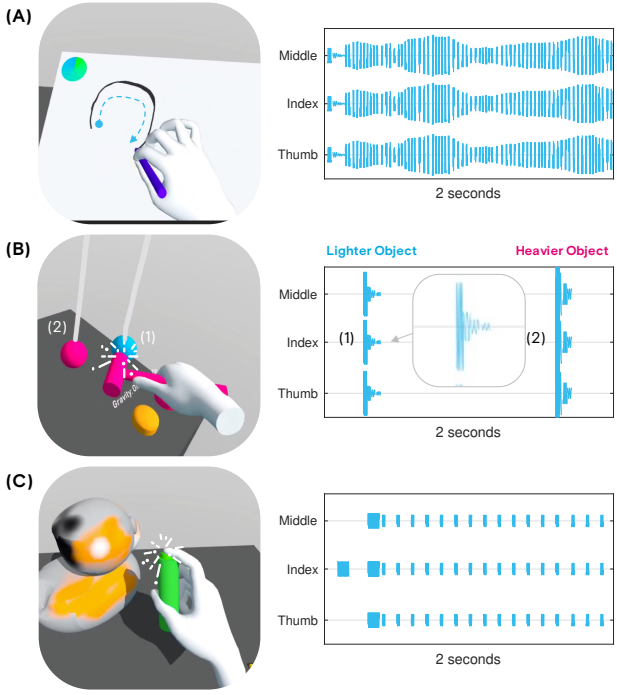

Figure 5.7: Haptic rendering output, shown as waveforms, during a time window of tool interaction in the following scenarios: (A) when drawing with the pen tool, (B) when striking objects of different weights with the hammer tool, (C) when spraying paint using the spray painting tool.

5.3 User Study

In this section, we introduce a sketching user study to evaluate the feasibility of using Haplets in a productivity scenario. We chose a sketching task because it encompasses the main concepts introduced in this paper: 1) a physical tool (a pen) is held by the user, thus allowing Haplets to augment the tool with vibrotactile haptics, 2) upon the pen contacting with a surface and while drawing, Haplets renders impacts and textures, and 3) physics-driven hands and tools respond to the sketching environment, introducing pseudo-haptic force and friction. The main task is loosely based on VRSketchPen [29] where the user would trace a shape shown on a flat canvas. With a simulated pressure sensitive pen, users would need to maintain a precise distance from the canvas in order to draw a line that matches the line thickness of the provided guide. We hypothesize that with Haplets providing vibrotactile feedback, users would be able to draw lines closer to the target thickness. In addition to the sketching task, after the end of the session, the user is presented with a manipulation sandbox for them to explore the remaining modalities that Haplets has to offer such as texture discrimination, object manipulation, and pseudo-haptic weight. The details for this sandbox is described in section 5.4.1.

5.3.1 Experimental Setup

We recruited 8 right-handed participants (2 females, aged 22–46, mean = 32.75, SD = 7.44) to participate in the study. Proper social distancing and proactive disinfection according to local guidance was maintained at all times and the study was mostly self-guided through prompts in the VR environment. The study was approved by our institution's IRB and participants gave informed consent. Participants were first seated and started by donning three Haplets on the thumb, index and middle fingers of their right hand. Then they donned an Oculus Quest 2 headset. Participants then picked up a physical pen and held it in their left hand using the headset's AR Passthrough mode, then a standalone VR Unity application was launched. All experiments were run and logged locally on the headset. All interactions

in VR were done using on-device hand tracking [41].

The VR environment consists of a single desktop with a large canvas, as shown in Figure 5.8. The canvas is used to present instructions, questions and the actual tracing task. Participants were first instructed to hold the physical pen in their right hand, which will also create a virtual pen in their hand using the tool detection system presented in the previous section. The participant uses the pen to interact with most elements in the environment including pressing buttons and drawing.

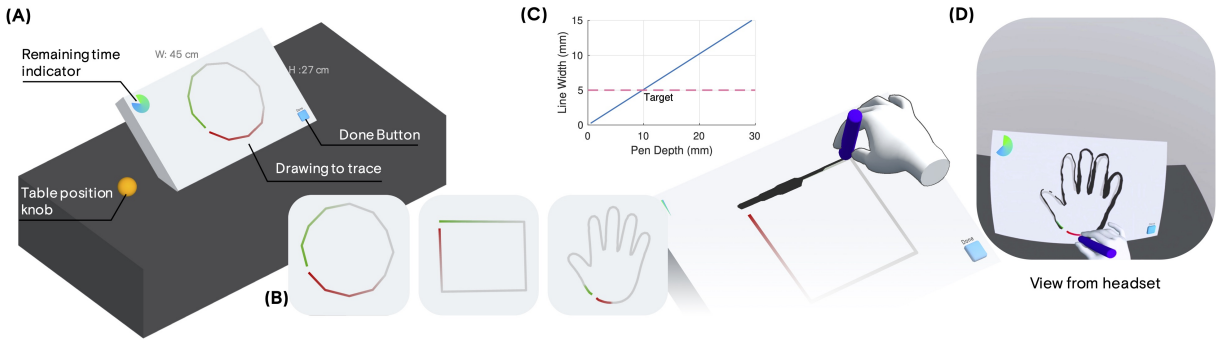

Figure 5.8: An overview of the user study environment: (A) An adjustable floating desk is presented to the user along with a canvas. The canvas contains template for the user to trace with along with an indicator for the remaining time in each repetition. The canvas can also present buttons for the user to indicate that they're done with the drawing or answers to questions. (B) Three shapes are used in the user study: a circle, a square and an outline of a hand. Participants start tracing at the green line section and end at the red line section. (C) A plot of the line width that results from how deep the users actual hand is penetrating the surface of the canvas. Participants must aim for the 5 mm line width which corresponds to a 10 mm depth. Vibrotactile feedback is provided as shown in Figure 5.7A. (D) An example of the participant's view captured from the headset during a trial.

The main task consists of participants tracing three shapes with the virtual pen: a square,

a circle and a hand, as shown in Figure 5.8. The shapes are presented in a randomized order and each shape is given 15 seconds to complete. The virtual pen is pressure sensitive and the lines the users draw vary in thickness depending on how hard the user is pressing against the canvas — we simulate this using our physically simulated hands, which means that the further the user's real hands interpenetrates the canvas, the thicker the line will be. The target thickness for all shapes is 5 mm. Lines are rendered in VR using the Shapes real-time vector library [6].

When drawing, for every 2.5 mm the pen has traveled on the canvas, Haplets render a vibration for 10 ms at 170 Hz with an amplitude that is mapped to how much virtual force is exerted on the canvas. Since the amount of force is also proportional to the line width, the amplitude of vibration is also mapped to the line width, as shown in Figure 5.8C. In other words, the pen and Haplets emulates the sensation of drawing on a rough surface and the pressure is represented as the strength of vibration. An example of the haptic rendering scheme's output is shown in Figure 5.7A.

Participants first perform 12 trials in a training phase to get familiar with the task where they would hold a physical pen but would not receive haptic feedback. Participants then performed 2 sets of 36 trials (12 of each shape per set) with or without holding the physical pen in their hand, totaling 72 trials. We balance the order of this throughout our participants to account for any order effects. Half of the 36 trials in each set have the Haplets turned off, presented in a randomized order. In summary, participants are given 2 conditions for either holding or not holding the pen and 2 conditions for either having or not having haptic feedback. After each set of trials, they are presented with a short questionnaire. After all trials have concluded, the participant is given a short demo of other capabilities of Haplets, as described in the following sections.

For each trial, we collected the line thickness for every line segment, the coordinates along each line segment, and the time it took to complete the drawing. Through post-processing,

---

[6]Shapes by Freya Holmér `https://acegikmo.com/shapes`

we calculate the mean line thickness for each trial, the mean drawing speed and the mean 2D error from the given guide.

5.3.2 Results

We performed a two-way repeated measures analysis of variance (ANOVA) for each independent variable: line thickness, drawing speed and drawing error with two within-subject factors: with or without haptic feedback, and with or without a physical pen held. Our analysis, as presented in Figure 5.9, revealed main effects for haptic feedback for line thickness ($F(1,7)=15.82$, $p<0.01$), drawing speed ($F(1,7)=10.75$, $p<0.02$), and drawing error ($F(1,7)=14.24$, $p<0.01$), but non-significance for the physical pen conditions nor the interactions between factors. Pairwise t-tests between the haptic and non-haptic conditions for each independent variable confirmed significance in line thickness ($p<0.001$), drawing speed ($p<0.001$), and drawing error ($p=0.001$).

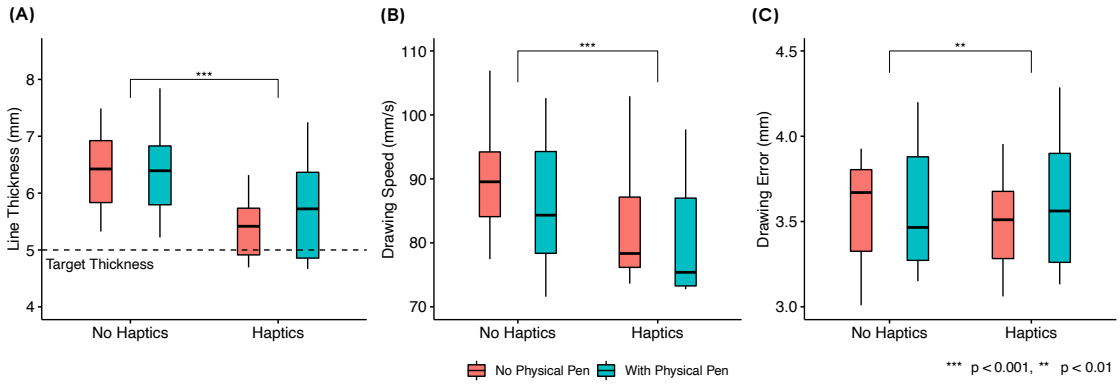

Figure 5.9: Results from the user study: (A) Participants draw lines that are closer to the target thickness (5 mm) with haptic feedback. (B) Users slow down significantly when haptic feedback is provided. (C) Users produce less 2D error when drawing with haptic feedback.

For line thickness, we can observe that participants can rely on the haptic feedback for guidance and draw lines that are closer to the guide's thickness, with an mean error across

subjects of 1.58 mm with haptic feedback and 2.68 mm without haptic feedback. Having a physical pen in the hand seems to negatively impact the line thickness, we attribute this to the deteriorated tracking accuracy when the physical pen occludes parts of the tracked fingers — a limitation of our particular setup. We confirmed this by observing video recordings of the sessions, where we could correspond moments of large line width variations to temporary losses of hand tracking (indicated by malformed hand rendering or hand disappearance). We can also observe that participants slow down significantly when haptic feedback is provided. We hypothesize that participants were actively using the haptic feedback to guide their strokes. The reduced drawing error is most likely to be influenced directly by the reduced speed of drawing.

When interviewed during a debriefing session after the experiment, two participants (P3 & P5) noted that they "forgot [the Haplets] were there". Some participants (P1, P3 & P8) preferred having the physical pen in their hands, while other participants (P2 & P7) did not. One participant (P7) suggests that they were more used to using smaller styli and thus preferred not having the larger pen when using a virtual canvas. Another participant (P6) noted that they felt the presence of the physical pen even after it has been removed from their hands.

From our questionnaire (Figure 5.10), we can observe that participants had a reasonable amount of body ownership (Q1) and agency (Q2, Q3). Having a physical pen in their hands did not seem to alter the experience of drawing on a virtual canvas (Q4, Q7). However, the presence of holding a virtual pen in VR seems to be positively impacted by having active haptic rendering (from the Haplets) along with the passive haptic feedback from holding a physical pen (Q5, Q6).

5.4 Demonstration

We built demonstrations for highlighting two potential use-cases for Haplets: 1. manipulation in AR/VR and 2. a painting application that uses our tool system.

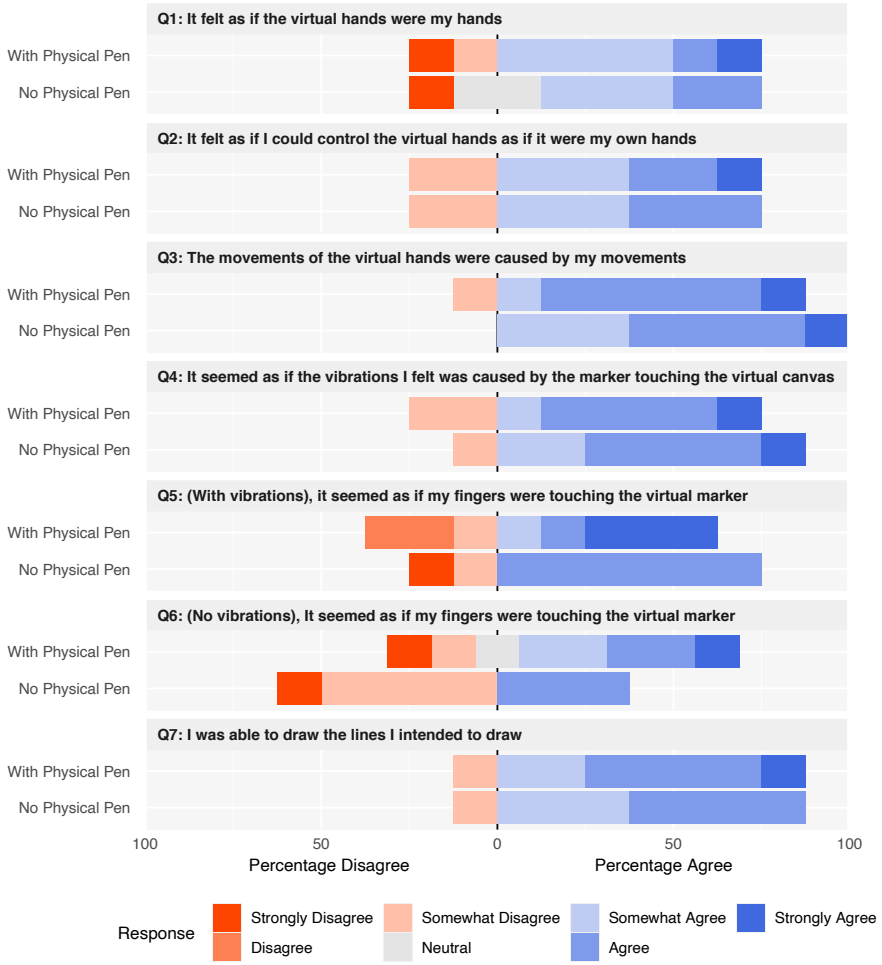

Figure 5.10: Results from the questionnaire provided during the end of each set of trials

5.4.1 Manipulation Sandbox

The manipulation sandbox is a demonstration presented to participants at the end of our user study. The user is presented with a desk with several objects and widgets, as shown in Figure 5.11.

The top-left corner of the desk consists of two squares with two different textures. The left square represents a smooth, black, plastic surface and the right square represents a grainy,

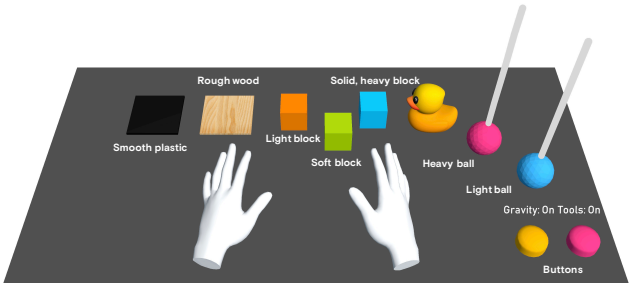

Figure 5.11: After the user study concludes, participants are presented with a desk with various objects to interact with. From left to right: a smooth plastic square, a rough wooden square, a lightweight orange block, a soft green block, a heavy blue block, a rubber ducky (solid), a heavy tethered ball, a lightweight tethered ball. Participants can use the lower right buttons to toggle the use of tools and toggle gravity on and off. Each object responds with haptic feedback as shown in Figure 5.5 and 5.7.

wooden surface. The smooth surface has a low coefficient of friction of 0.1 while the rough surface has a high coefficient of friction of 1.0. When the user runs their fingers across each surface, the haptics system renders different frequencies for each texture, at 200 Hz and 100 Hz, respectively. Each vibration is generated after the fingertips have traveled at least 2.5 mm, similarly to the pen's haptic rendering scheme presented in the previous section. Since the wooden texture has higher friction, the user's finger will stick and slip, rendering both visually and through haptics, the sensation of a rough surface. An example of the surfaces' haptic rendering system in use is shown in Figure 5.5A & 5.5B.

Towards the center of the desk are three cubes with varying densities. The user can either pick up or prod the cubes to figure out which cube is lighter or heavier than the others. When prodded at, visually, the lighter cube will slide while and heavier cube will topple. When picked up, the lighter cube will render a lower control-display ratio (pseudo-haptic weight illusion) than the heavier cubes. Each cube also responds to touch differently. The lighter cube will render a vibration of 100 Hz upon touch, to simulate a softer texture, while the

heavier cube will render a vibration of 200 Hz to simulate contact with a dense object. Held cubes can also be tapped against the desk, and similar vibrations will be rendered upon impact. Upon release, Haplets will render a smaller amplitude vibration of the same frequency. A rubber duck is also presented nearby, with similar properties to the cubes. An example of the haptic rendering output is shown in Figure 5.5C.

Towards the right of the desk are two buttons: one for toggling gravity on and off and another for toggling the tool system on and off. The button responds to initial touch using the same system as other objects but emit a sharp click (170 Hz, 20 ms) when depressed a certain amount to signal that the button has activated. If the user turns the tool system off, tools will not be created when a pose is recognized. The buttons' haptic rendering scheme is shown in Figure 5.5D.

When the tool system is active, users can create a hammer in their hand by holding the "thumbs-up" pose (see Figure 5.6). The hammer can be used to tap and knock the items on the desk around. The hammer's haptic system is similar to the fingertips, where each object responds with different frequencies depending on pre-set properties and different amplitudes depending on how much reaction force is generated when the hammer strikes the object. All three Haplets will vibrate upon hammer strikes under the assumption that the user is holding the hammer with all three fingers. Additionally, a lower frequency reverberation is also rendered immediately after the initial impact in order to emulate stiffness. An example of the haptic rendering scheme for the hammer is shown in Figure 5.7B.

5.4.2 Painting

Our painting demonstration, as shown in Figure 5.12, is designed to highlight the use of our tool system. A desk with a large, gray duck sculpture is presented to the user. The sculpture can be rotated using the user's bare hands. The user can use three tools during the demonstration: a spray bottle, a pen and a hammer. Each tool is placed in the user's hand when they produce the correct pose, as shown in Figure 5.6. The lower right corner of the desk contains a palette of five colors: the user can tap the tool on the color to switch

the tool to operate with that color.

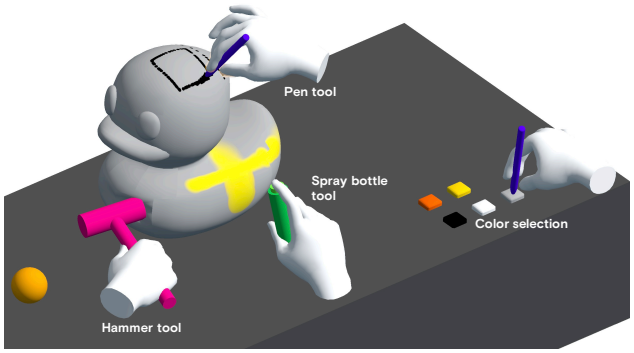

Figure 5.12: The painting application highlights the use of haptic feedback to enhance the experience of using otherwise passive tools in the hand. Shown are four interactions overlaid from left to right: using the hammer tool to adjust the sculpture's geometry, using the pen tool to draw on the sculpture, using the spray bottle to spray paint on the sculpture, and selecting colors by tapping the tools on the swatches provided. Haptic feedback provided by the tools is visualized in Figure 5.7.

The spray bottle is used to quickly paint the duck sculpture. As the user presses down on the bottle's nozzle, a small click is rendered on the index finger Haplet. When the nozzle is engaged and the tool is producing paint, all three Haplets pulse periodically with an amplitude that corresponds to how much the nozzle is depressed. An example of the haptic rendering output is shown in Figure 5.7C. Paint is deposited onto the sculpture similar in behavior to spray painting.

The pen is used to mark fine lines on the sculpture. The haptic rendering scheme is identical to that of the pen described in the user study, where all three Haplets render vibrations with amplitudes that correspond to the depth of penetration and line width.

The hammer is used to modify the sculpture by creating indentations. The sculpture's mesh is modified in response to the reaction force caused by strikes of the hammer. Haptic

feedback for the hammer is similar to that presented in the previous section (Figure 5.7B).

5.5 Discussion

Haplets introduces a wireless, finger-mounted haptic display for for AR and VR that leaves the user's fingertip free to interact with real-world objects, while providing responsive vibrotactile haptic rendering. Each Haplet is a self-contained unit with a footprint small enough to fit on the back of the fingers and fingernail. We also present an engineering solution to achieve low-latency wireless communication that adds haptic rendering to various use cases in VR such as manipulation, texture rendering, and tool usage. Our simulation system for physics-driven virtual hands complements our haptic rendering system by providing pseudo-haptics, robust manipulation, and realistic friction. With a real-world tool held in the hand, Haplets render vibrotactile feedback along with visuals from our simulation system to augment the presence of the tool. Our user study and demonstrations show that Haplets is a feasible solution for a low-encumbrance haptic device.

Although Haplets exclusively provide vibrotactile feedback, we have introduced several engineering efforts to maximize the rendering capabilities of our haptic actuator, the LRA. Our brief characterization of the LRA shows that LRAs can be used at frequencies outside the resonant frequency when properly compensated for. Furthermore, our characterization also shows that the material (or body part) on which the actuator is mounted on to can cause the output of the actuator to vary significantly and therefore needs to be characterized for the intended location of the actuator. Our low-latency wireless communication also helps minimize the total latency from visual stimuli to tactile stimuli, which is especially useful when considering the inherent mechanical time delay for LRAs.

Our user study and subjective feedback from the demonstrations have shown that Haplets and the current framework do provide adequate haptic feedback for the given tasks and experiences. However, the human hand can sense much more than simple vibrations such as the sensation of shear force, normal force, and temperature. We address this shortcoming by introducing believable visuals in the form of physics-driven hands, and make up for the

lack of force rendering by introducing passive haptic feedback in the form of tools. Our low-latency solution enables impacts, touch and textures to be rendered responsively according to the simulation and visuals. Furthermore, we have yet to fully explore the voice coil-like rendering capabilities of the haptic actuator (LRA). Therefore, our immediate future improvement to our system is the ability to directly stream waveform data to the device. This will enable the ability to render arbitrary waveforms on the LRAs or VCMs which can be used to render highly realistic textures [12] or the use of audio-based tools for authoring haptic effects [54, 99].

Our implementation of passive haptic feedback for tools uses a pen for physical grounding of the fingers, which provides adequate grounding for a number of tasks. Inertial cues that provide the sense of weight to the pen are presented using pseudo-haptic weight. [127] have shown that VR controllers with reconfigurable shapes can provide realistic haptic cues for inertia and grounding. Therefore, a potential venue for future work is the use of Haplets in conjunction with actual tools (e.g. holding an actual hammer or an actual spray can) or reconfigurable controllers, which would not only provide realistic grips and inertia but also additional haptic feedback that the tool may provide such as depressing the nozzle of a spray bottle.

For other potential future work, our framework provides a foundation for building wearable haptic devices which are not necessarily limited to the fingers. In its current form, Haplets can be placed on other parts of the body with minimal adjustments for rapid prototyping haptic devices, such as the forearm, temple, and thighs [22, 129, 118, 98]. With some modifications, namely to the number of motor drivers and firmware, Haplets can also be used for rendering a larger number of vibrotactile actuators at once, which could be used to create haptic displays around the wrist or on the forearm. Furthermore, our motor drivers are not limited to driving vibrotactile actuators, skin stretch and normal force can be rendered with additional hardware and DC motors [106].

5.6 Conclusion

We have introduced Haplets as a wearable haptic device for fingers in VR that is low encumbrance. Haplets can augment the presence of virtual hands in VR and we strengthen that further with physics-driven hands that respond to the virtual environment. We have also introduced an engineering solution for achieving low-latency wireless haptic rendering. Our user study and demonstration shows that Haplets have potential in improving hand and tool-based VR experiences. Our system as a whole provides a framework for prototyping haptic experiences in AR and VR and our immediate future work is exploring more use cases for Haplets.

Chapter 6

ACTUALLY ACTIVE ELECTRODES: PNEUMATICALLY ACTUATED ELECTRODES FOR EMG-BASED INTERACTION IN VIRTUAL REALITY

Pornthep Preechayasomboon[1], Eric Rombokas[1]

1 Mechanical Engineering, University of Washington

6.1 Introduction

Surface electromyography (surface EMG or sEMG) has long been used as a measure of the intent of the underlying muscles in the limb. Their most prominent use has been for medical diagnostics and prosthetics [86]. More recently, surface EMG has shown potential to becoming used as an input modality [60], or at least a supplement to other input modalities, for virtual reality (VR), augmented reality (AR) [72], and general human-machine interaction [130] or human-computer interaction [43]. For AR and VR, most have targeted decoding the actual pose of the hand from EMG on the wrist or forearm [77, 84], while others incorporate force exertion [13, 68, 117] or intent [115, 28].

Using hands is one of the most intuitive input modalities for AR and VR as hands are one of the most dexterous tools humans naturally possess. Tracking the pose of the user's hand, also referred to as hand-tracking, is most often used as the form of input where a virtual hand closely resembling the current pose of the hand is used to interact with elements in the virtual environment [41]. The virtual hands can then be used to manipulate objects, gesture for communication, or interact with interfaces. Hand-tracking is often coupled with other supporting systems such as pinch-based input [64], physics simulation [8], haptics [145, 30], and pseudo-haptic force illusions (rendering the hand offset from the actual position depending on force) [2, 120] to help enhance the subjective realism of the virtual environment.

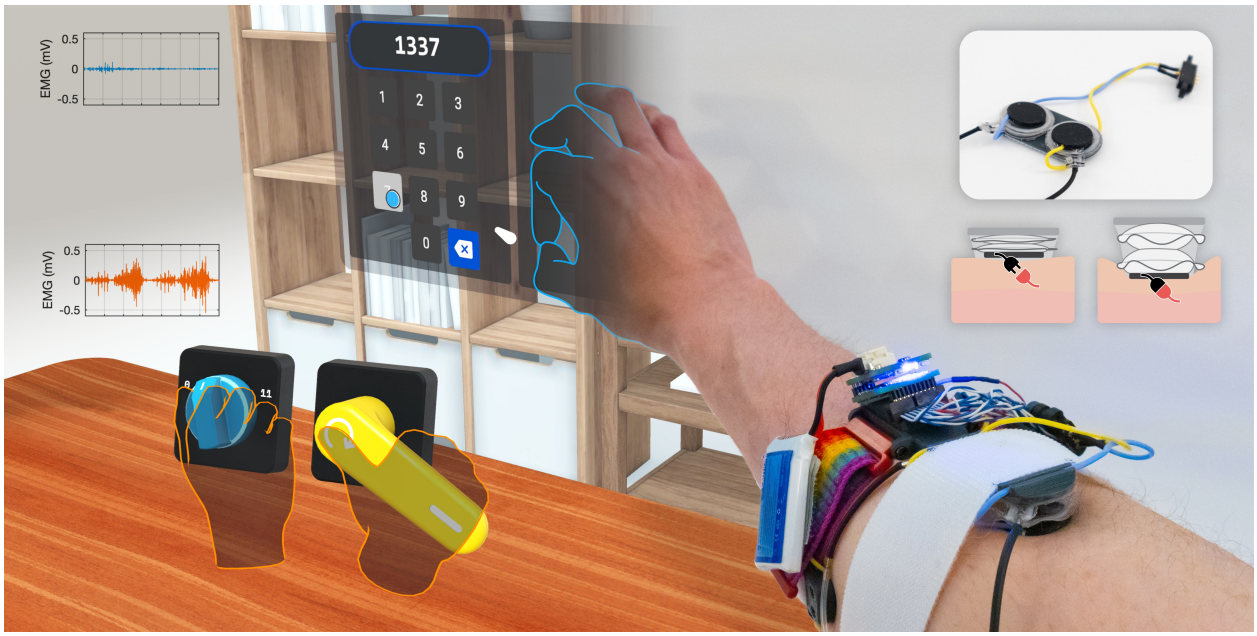

Figure 6.1: As we use electromyography (EMG) on the forearm as input for virtual reality, there is a chance that the electrodes conducting the minute signals from the skin to the device can lose contact with the skin. Our device (right), presents a solution by actively monitoring and actuating the electrodes to maintain good contact with the skin.

Hand tracking can be achieved in multiple ways. At the time of writing, the most prominent method is optical, where the pose of hands are extracted from images from cameras mounted either on an AR or VR headset [41, 146], a smartphone [38, 7], or a stationary device [145]. Some methods require multiple infrared cameras [41], while others require monocular RGB images [38] with a varying amount of accuracy between methods. Other methods include glove-based tracking [152, 42], magnetic tracking [14], and abstractions of hand-tracking such as finger tracking from a ring-based device [96]. As mentioned previously, sEMG has also shown promise to achieve hand-tracking. Research that targets some semblance of the hand such as detecting static gestures [55], or force exertion [82] rely either on laboratory-grade or medical-grade equipment for achieving the highest quality EMG signal possible at the cost of bulkiness, or wearable systems such as the Myo Armband or Myo Armband-like systems that trade signal resolution for long-term wearability [131]. With the latter being more desirable for consumer use.

The reliance on high signal quality is mainly due to challenges in signal processing as sEMG signals are miniscule (in the order of millivolts) and prone to external interference and motion artifacts [86]. In the case of laboratory-grade equipment for signal acquisition, surface EMG can rely on wet gel electrodes that conduct the minute electrical signals from the skin to a signal amplifier. These gel electrodes often have adhesives to help maintain the position and contact of the electrode on the skin. Using dry electrodes is another means of signal acquisition, which is more preferable due to not requiring gel to aid in signal conduction. However, dry electrodes may require skin preparation, and have more susceptibility to motion artifacts and interference artifacts [153]. These issues are exacerbated when dry electrodes are part of a wearable device, especially those without adhesives, which have little to no means of correcting for the loss of contact with the skin. Recent advances in materials, textiles, signal acquisition methods and soft robotics, have shown improvements in achieving high signal quality for EMG [93]. Yet, aside from potentially warning the user of poor contact [50], these dry electrode-based devices have no means of self-correction.

Wearable devices often require flexibility and conformability while maintaining function-

ality, including wearable devices for sEMG sensing. Recent advances in soft robotics, has shown that wearable device design can benefit from soft robotic design, fabrication, sensing and actuation. Most notably, soft robotic actuators have been integrated in soft wearable haptic displays. Some rely on pneumatic actuation to render local pressure [155, 3], while others rely on electrochemical liquid-gas phase transitions [147] or dielectric elastomers [158]. These devices show potential in the ability to actuate elements close to the skin, all while being soft and conformable to the body's geometry. Although not commonly studied, some haptic displays in research have elements that are designed to adjust to achieve better display performance. One example, PneuSleeve [163], has multiple pneumatic elements that equalize grounding force when rendering shear forces and user-specific grounding pressures.

In this chapter, we integrate soft robotics into an sEMG system: we present a pneumatic system that can dynamically adjust the coupling pressure between the surface EMG electrode and the user's skin to maintain adequate electrode-skin contact for use as a VR input device. Our system is comprised of a pneumatic band based on soft robotic bellows actuators and an EMG armband with dry electrodes. We hypothesize that while using an EMG-based input modality for VR, the electrical coupling between the EMG electrodes and skin directly effects the performance of the user and that maintaining a good coupling positively effects performance. We detail the design, fabrication and characterization of all major elements of our system. We integrate our system into a virtual reality application for studying how good and poor electrode contact effects EMG-based input. Lastly, we present a proof-of-concept prototype of a closed-loop system to maintain adequate electrode contact for EMG using soft robotic actuators.

We present the following contributions:

1. The design and fabrication of a wireless 8-channel surface EMG device for data collection and input for AR and VR

2. The fabrication and characterization of novel 3D-printed surface EMG dry electrodes using consumer-grade 3D printers

3. A novel fabrication process for pneumatic bellows using consumer-grade 3D printers

4. The design and fabrication of a pneumatic forearm-band for dynamically adjusting the coupling pressure between dry electrodes and skin using bellows actuators

5. EMG-based modulation for input selection between physical objects and panel interfaces in VR

6. A study on the effects of good and poor electrode contact with the skin on sEMG-based input for VR

7. A dataset of sEMG and hand pose recordings on various conditions of electrode-skin coupling pressure and activities in VR

8. A proof-of-concept closed-loop soft robotic system that maintains a good electrode-skin coupling using sEMG readings as input

6.2 Design

In this section, we describe the design and fabrication of several core elements of our system: a wireless EMG data acquisition system, 3D-printed dry electrodes for sEMG sensing, pneumatic bellows actuators and a microblower-based pneumatic system. We combine these elements into a wearable sEMG armband and wearable pneumatic band that are used together to study the effects of varying the skin-electrode coupling pressure on sEMG sensing.

6.2.1 EMG System

For the purpose of data collection and use in a real-time system, we developed our own modular EMG device with 8 bipolar channels based on an off-the-shelf analog front-end (AFE), the ADS1299 (Texas Instruments). The overall design of the electrical circuit is a derivative of the OpenBCI Cyton[1] and is miniaturized to a 19 by 22 mm footprint. We

[1] `https://github.com/OpenBCI/V3_Hardware_Design_Files`

pair the ADS1299 with an nRF52832 (Nordic Semiconductor) system-on-a-chip (SoC) for wireless communication. The EMG device can stream 24-bit 8 channel raw analog readings from the ADS1299 at 1000 Hz to a host device using the wireless Enhanced Shock Burst (ESB) protocol of the nRF52 SoCs. Our host device, an nRF52840-based USB dongle (Nordic Semiconductor), can then forward the communication to and from USB connected hardware such as personal computer running macOS, Windows, Linux or Android, including virtual and augmented reality headsets like the Quest 2 (Meta). The analog electrical interfaces of the ADS1299 are exposed via a miniature 20-pin 1.27 mm pitch dual-row header which enables swapping between various prototypes and form-factors.

3D-Printed Dry Electrodes for sEMG

As mentioned previously, dry electrodes are preferred over wet electrodes for long-term or everyday use—and we desire to use dry electrodes for our system as well. Most commercially available dry EMG electrodes are not designed for integration into other systems, therefore, we fabricated our own dry electrodes by 3D-printing electrically conductive carbon-black infused thermoplastic polyurethane (TPU), specifically PI-ETPU 95-250 Carbon Black (Palmiga Innovation). PI-ETPU has a Shore A hardness of 95 and is flexible and slightly elastic. Our round electrodes are 14 mm in diameter and 1.5 mm thick. We embed the ends of thin electrical wires into the electrodes during printing by pausing printing at a specific layer height and placing the wires in a small channel, the 3D printer then lays material above the channel thus entrapping the wire end in the electrode, as shown in Figure 6.2. These wires are used to connect the electrodes to our EMG system's ADS1299 AFE.

EMG Armband

We combined our EMG system and 3D-printed electrodes into a wearable armband prototype. The EMG armband is constructed from a slightly modified nylon smartwatch strap where the electrodes are attached onto 3D-printed nodes along the circumference of the strap (see Figure 6.3). Every channels' electrode is wired directly to the EMG system which is

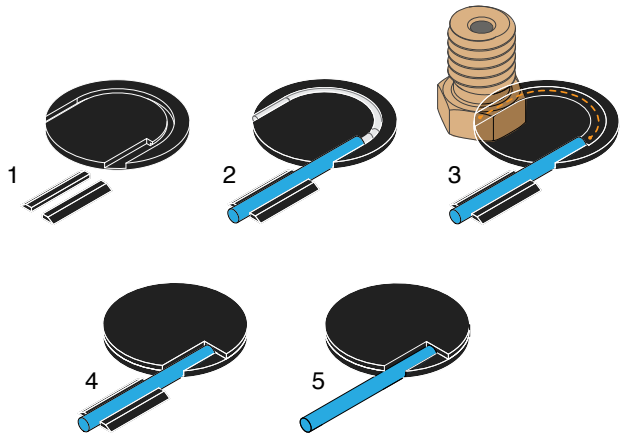

Figure 6.2: Dry electrode fabrication process: 1) Electrically conductive TPU is 3D-printed in the shape of an electrode with a small channel for a stripped wire to be placed into. Two small guides are also printed to help align the wire's insulation. The printer is paused. 2) A 30 AWG stranded wire with silicone insulation is placed and pressed into the channel manually until flush with the top surface. 3) The printer resumes and electrically conductive TPU is extruded into channel to seal the stripped portion of the wire in place. 4) Two additional layers of electrically conductive TPU is printed to form the top surface of the electrode. 5) The electrode is completed and the guides can be discarded. Adhesive can be added to secure the insulation of the wire to the 3D printed material.

attached atop one of the nodes. Once the armband is worn on the forearm, a small 3.7V, 380 mAh Lithium-Polymer battery can be connected and secured onto the band's hook-and-loop fabric. In our particular setup, one channel is left exposed as a magnetic connector for connection to a pair of electrodes external to the armband. The armband and battery combined weighs 61 grams, and is approximately 38 mm wide.

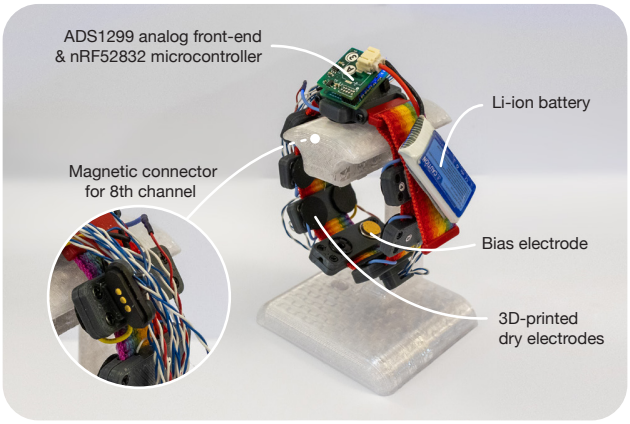

Figure 6.3: Our EMG armband prototype. The armband is primarily constructed from a nylon smartwatch strap fitted with 3D-printed housings for our 3D-printed dry electrodes. 7 channels are positioned along the length of the band with 1 remaining channel (channel 8) exposed as a magnetic connector. A gold-plated bias electrode is used driving an inverted common-mode signal comprised of the average from all channels to aid in the reduction of power-line interference [50].

6.2.2 Actuated Electrodes

Bellows Actuators

We use inflatable bellows actuators (also referred to as just "bellows") that expand with air pressure in order to vary the amount of pressure acting on the EMG electrodes on the forearm. The bellows are a derivative of the actuators from BellowBand [155]. We developed

a process to manufacture bellows on a consumer-grade 3D printer—without the need for an industrial heat press or intermediate paper layers. Like BellowBand, the bellows are layers of thermoplastic polyurethane (TPU) sheet that are sealed together to create expandable structures. To produce each bellows, we first cut patterns into a TPU sheet using a desktop cutting machine (Silhouette Cameo 4, Silhouette America) to create individual layers. Then, the first layer is glued onto the bed of a 3D printer (Prusa MINI, Prusa Resaerch) followed by placing the next layer on top. The 3D printer then extrudes TPU material (FlexFill TPU 98A, Fillamentum) to seal the layers together in predefined patterns akin to the heat-pressed patterns as presented in [155]. The process is then repeated until the stack-up of layers is complete. The TPU sheets are then removed from the bed to be post-processed, where a short 1.1 mm diameter copper tube is inserted into the bellow's opening and sealed with ultraviolet-cured cyanoacrylate adhesive (Loctite 4310, Henkel). The entire process is also illustrated in detail in Figure 6.5. Each bellows actuator is 20 mm in diameter and approximately 1 mm thick when not inflated. We attach our EMG electrodes to each bellows actuator using double-sided adhesive as shown in Figure 6.10. A larger, 50 mm wide, variation of the bellows actuator design is shown in Figure 6.7 and Figure 6.10.

Pneumatic Microblower System

Pneumatic systems often rely on large components such as industrial pumps, valves and manifolds which is often prohibitive for wearable systems. More recently, projects such as FlowIO [128] have provided open-source, battery-operated pneumatic systems that are compact enough to be wearable. However, at the time of writing, their system is a miniaturization of the aforementioned components—the pumps, manifolds and valves are still present, only in a smaller form-factor. Piezoelectric-based microblowers have recently been used in wearable blood pressure monitoring devices such as the HeartGuide (OMRON Healthcare). Here, we developed a pneumatic system based on piezoelectric microblowers that is designed for general purpose use and an operating range suitable for our bellows actuator.

We use MZB series piezoelectric microblowers (MZB3004T04, Murata Manufacturing)

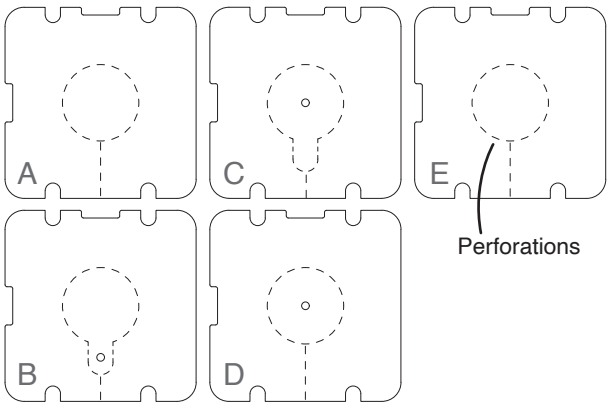

Figure 6.4: The cut patterns for the 0.15 mm thick TPU sheets is shown. Each pattern has notches for alignment with a 3D printed guide during fabrication. Perforations are cut into the sheet in the shape of the final dimensions. After fabrication on the 3D printer, the patterns can be torn manually along the perforations to achieve the final shape.

for our system. We chose the MZB3004T04 due to its availability from major suppliers and pressure output envelope. The MZB3004T04 is approximately 19 mm by 21 mm wide and 3.4 mm tall excluding the outlet port. The microblower generates pressure by vibrating a piezoelectric element at ultrasonic frequencies (21.5 - 24.5 kHz) to push air through an internal one-way diaphragm, thus building up pressure beyond the diaphragm. The outlet port can be readily connected to 3 mm inner diameter tubing. As we desired to tightly control the output pressure of the microblower, we designed and fabricated a 3D-printed adapter for the outlet port to house a micro pressure sensor (MPR series, Honeywell) and a smaller 1.25 mm outlet port for 1 mm inner diameter tubing, as shown in Figure 6.6.

Controller

As each microblower is intended to be modular, we designed a controller circuit board that can be attached on the flat inlet side of the microblower, as shown in Figure 6.6. The controller consists of a full-bridge piezo driver (LT3572, Linear Technologies) capable of

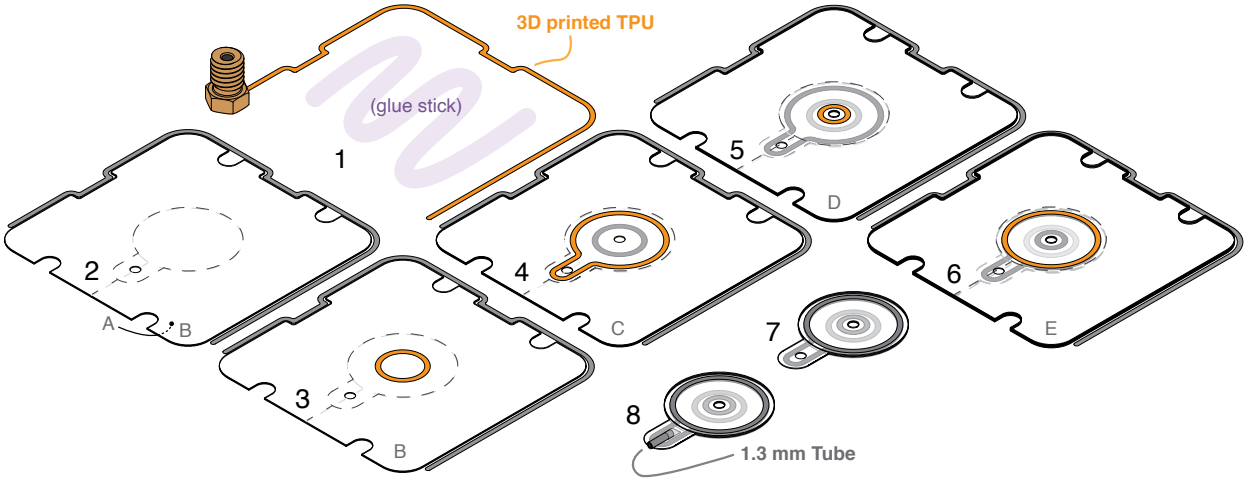

Figure 6.5: The steps for fabricating bellows actuator using the cut TPU patterns from Figure 6.4 is shown. 1) A guide is printed on the print surface with notches to align the TPU patterns. Adhesive from a glue stick is manually deposited onto the print surface inside the guide. 2) Patterns A then B are manually placed on the print surface. 3) TPU is printed in the shape of a ring joining patterns A and B along the ring. This forms the base of the bellows actuator. 4) Pattern C is placed on top followed by TPU printing in the shape as shown to join pattern B and C which forms the inlet. 5) Pattern D is placed on top followed by TPU printing of a small ring joining pattern C and D. This completes the first bellows actuator and forms the base of the second bellow. 6) Pattern E is placed on top and the final ring of TPU is printed. This forms the second bellow. 7) The bellows are removed from the print surface and patterns are manually torn along the perforations. 7) A 1.3 mm diameter copper tube is inserted into the 2 mm opening at the inlet and a small bead of UV-cured adhesive is used to pneumatically seal the tube and bellows.

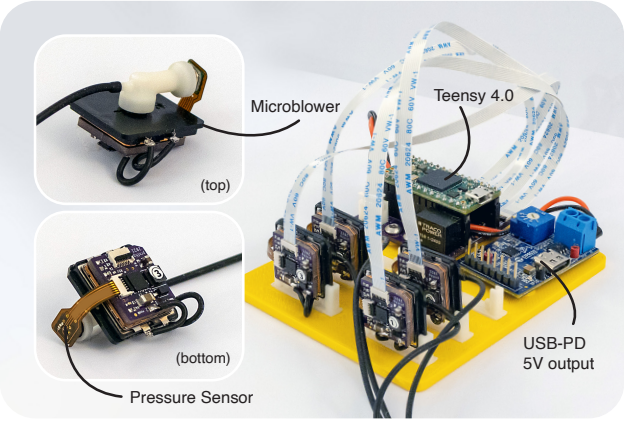

Figure 6.6: Our microblower pneumatic system consists of four piezoelectric MZB3004T04 microblowers, each with its own driver, and a Teensy 4.0 microcontroller. Power is supplied using a USB-PD supply at 5V that is regulated to 3.3V for the drivers' boost-converter.

driving the microblowers at their maximum voltage rating (19 V) with a built-in boost-converter supplied with 3.3V and pulse-width modulated (PWM) frequency at 21.5 - 24.5 kHz. A flex PCB is used to connect pressure sensor the controller board and the I2C and supply is passed through to a 8 pin flat flex cable (FFC) which houses both communication for the piezo driver, pressure sensor and their respective power supplies.

We connect 4 microblowers and their respective pressure sensors to a microcontroller (Teensy 4.1, PJRC) in order to control each microblowers' output pressure. In our use case, each microblower is connected to a single bellows actuator. We use a proportional-integral (PI) controller sampled at 200 Hz to control the PWM output of each bellows using pressure readings from the pressure sensor as input. Each microblower is tuned for its maximum output performance by setting the PWM frequency nearest to its resonance frequency (approximately 23$\pm$0.5 kHz). The PI controller varies the amount of output by varying the duty cycle of the PWM frequency. Since the microblowers operate using the vibrations generated from the PWM frequency, the usable duty cycle is from 0% to approximately 50% —increasing the duty cycle higher than 50% would start to reduce the

output instead until no output is generated at 100% as the piezoelectric element is no longer moving. The microcontroller can then be connected to a device to be controlled via USB HID, similar to our EMG band.

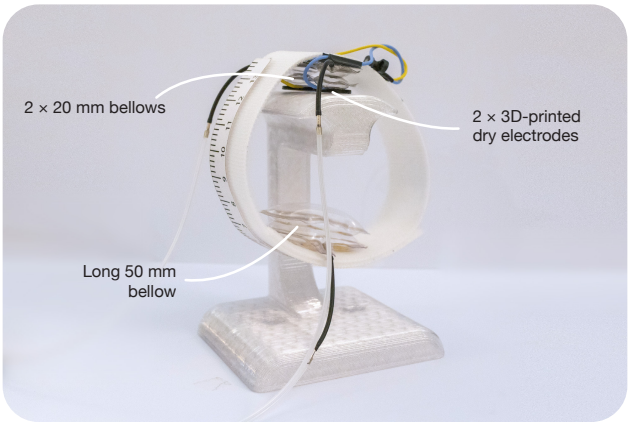

Figure 6.7: Our pneumatic band for varying the skin-electrode coupling pressure. The band is constructed primarily from a fabric hook and loop strap. Two bellows actuators with dry electrodes, forming one channel, is attached at the center of the band. A larger, 50 mm long bellow, is attached at one end of the band. Together, all 3 bellows actuators can tighten or release the pneumatic to varying degrees depending on the amount of pressure supplied.

6.3 Characterization

In this section we describe our characterization and validation efforts for two key components of our system: our 3D-printed dry electrodes and our pneumatic band and its system. Our efforts here serve as reference for future studies using our systems or subsystems.

6.3.1 EMG Electrodes

Using the fabrication method described in 6.2.1, we produced a total of 18 3D-printed dry electrodes which are used for characterization and subsequently in the prototype EMG armband used in the sections that follow. We first measured the resistance of the electrodes by

subjecting them to a constant normal force loading of 2, 4, and 8 N against a copper strip. The electrodes have a rough contact surface from the 3D-printing process as it is printed onto a rough textured sheet. The rough surface is intended to aid in maintaining contact good electrical contact with human skin by increasing static friction and increasing the overall surface area. However, when in contact with a smooth, hard surface we hypothesize that the amount of contact will vary with the amount of normal force applied thus leading to a decrease in resistance as electrical contact increases when force increases. Our measurements show that the electrical resistance of our electrodes vary with normal force applied, with a mean across all electrodes of 1.071 kΩ ± 513.75 Ω, 621.17 Ω ± 280.86 Ω, and 355.80 kΩ ± 118.20 Ω for 2, 4, and 8 N of normal force respectively.

Although our electrodes exhibit high variability in their resistance across different electrodes, the electrode-skin impedance has a higher effect on the signal quality [86] and is up to three orders of magnitude larger than electrode resistance. We performed two brief experiments to characterize our electrodes: 1) measuring and comparing the electrode skin-impedance across different conditions of loading and types of electrodes, and 2) measuring and comparing the measured EMG signal on adjacent locations on the forearm between our dry electrodes and standard wet gel electrodes

Electrode-Skin Impedance

We use a custom forearm pressing device that can hold a pair of electrodes underneath a carriage that can vary the amount of force being pressed on the forearm using a force sensor inline with the electrode (Figure 6.8). The user's forearm is positioned underneath the electrodes and the electrodes are connected to an impedance analyzer (Analog Discovery 2, Digilent Inc). The electrodes are then pressed onto the forearm until a stable force is held at 2, 4, 8, and 10 N for each repetition and the impedance is measured at 10 to 2000 Hz of frequency. The results (Figure 6.8) show that at 100 Hz, our dry electrodes on unprepared and dry skin have an electrode-skin impedance of 2433.8±275.05, 1742.54±255.32, 1428.36±106.17, and 1233.35±127.31 kΩ for 2, 4, 8, and 10 N of force. However, with a

small amount of sweat, which is simulated using a small amount of saline solution on the skin, the electrode-skin impedance reduced to 254.5±24.10 kΩ across all loading conditions. This small amount of sweat is akin to having dry electrodes placed on the skin for a long duration (> 1 hour) of time [58, 153]. It can be seen that the impedance is still well above the standard wet gel electrodes, but is in line with dry electrodes found in research. Note that this comparison should be considered anecdotal since the electrode-skin impedance can vary from person-to-person and conditions of the skin, however, the ADS1299 amplifier used in our EMG band has an input impedance of at least 1 GΩ, which our electrodes are well below the commonly maximum recommended impedance of 1/10 times the input impedance (lesser than 100 MΩ) [70].

Validation Against Commercial Wet Electrodes

Using commercially available gel electrodes (Red Dot 2560 Electrodes, 3M) as a baseline, we compared the EMG recordings from our dry electrodes on our EMG band. We had a user don the EMG band while connecting the eighth channel of our EMG band to a pair of gel electrodes, as shown in the top image in Figure 6.9. After a warm-up duration of 5 minutes to let skin conductance reach reasonable stability, we recorded 40 seconds of EMG signals. Here, we compare channel 4 from our EMG band to the gel electrode (channel 8) which are adjacent to each other on the forearm. Although not ideal, this gives us reasonable measurements of muscle activity along the same muscle (extensor digitorum). Figure 6.9A & B shows an 8 second excerpt of the collected signals where the signals, visually, have high similarity in both shape and envelope. Following [153], we report a Pearson's correlation of 0.97 between our dry electrodes and the gel electrode envelopes across the entire duration, and a signal-to-noise ratio (SNR) of 56.1 dB and 57.68 dB for our dry electrodes and the wet electrodes, respectively. A sliding correlation with a window of 1000 samples is shown in Figure 6.9C, as well as a Welch's power spectral density (PSD) plot (Figure 6.9D & E) of different muscle activities for visual comparison between the two electrodes. Note that the validation presented here is brief, and a complete validation that takes into account long

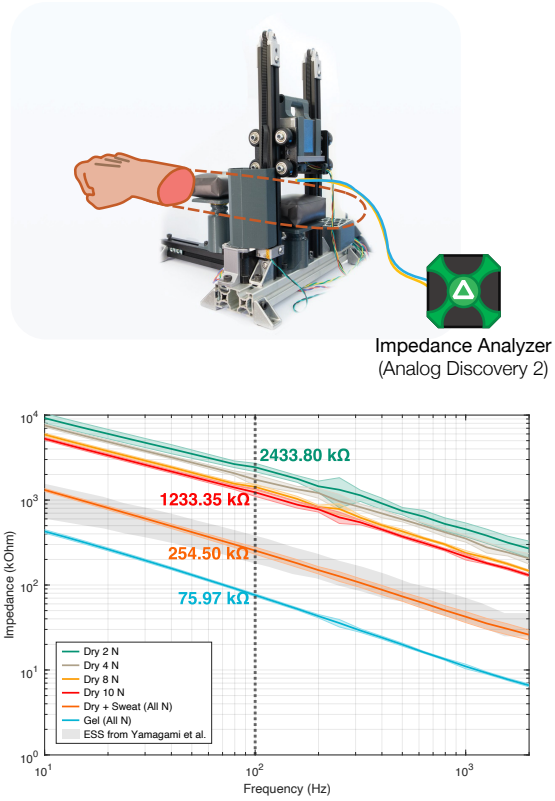

Figure 6.8: (Top) Our custom electrode testing device, where weights can be placed atop a carriage that holds a pair of electrodes with an inline force sensor. (Bottom) The skin-electrode impedance measurements from our brief dry electrode characterization. Impedance is plotted over frequencies of 10 to 2000 Hz, with the (transparent bands) showing standard deviation between measurements. It can be seen that for dry electrodes that have initially come into contact with skin, the amount of normal loading (force) can affect the impedance with lower forces giving rise to higher impedance. However, with a small amount of sweat formation, occurring approximately 5 minutes after skin contact, impedance is similar throughout all force levels. Gel 3M Red Dot electrode impedance is plotted for reference. Electrode impedance data from [153] is also included for comparison.

wear duration and drift over time is beyond the scope of this paper.

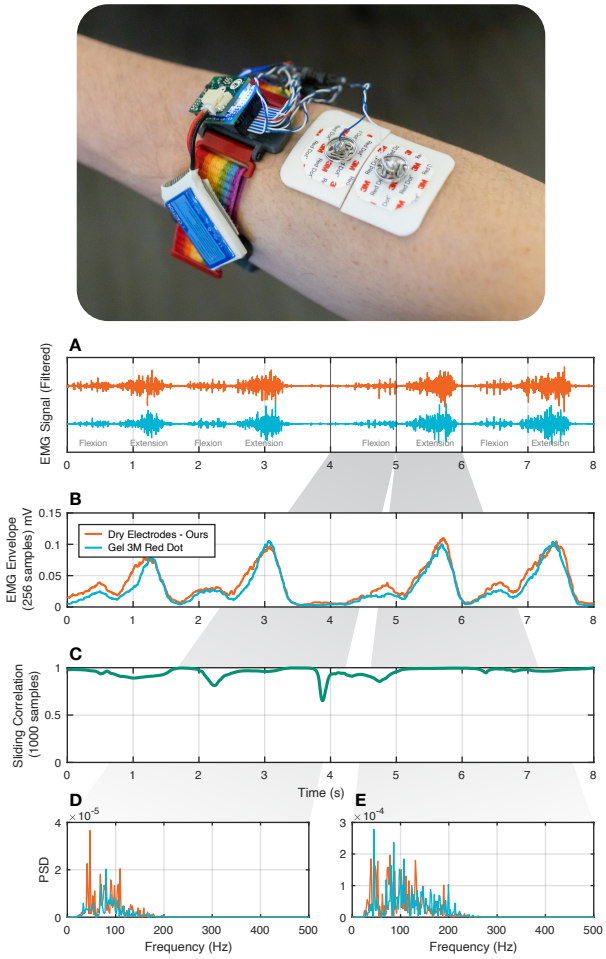

Figure 6.9: (Top) 3M Red Dot gel electrodes were connected to the eighth channel of our EMG band for comparison against the adjacent channel. (A) The filtered signal from both types of electrodes is shown, the overall magnitude and envelope of the signal is similar. (B) The linear envelope of the signals from both electrodes are similar as confirmed by (C); the envelopes are constantly highly correlated throughout the excerpt shown. (D) & (E) show the power spectral density (PSD) during wrist flexion and extension for both electrodes.

6.3.2 Bellows Actuators

We fabricated our bellows actuators in similar dimension and shape to [155], readers who desire an in-depth characterization using a more capable pneumatic system should refer to the cited paper. Here, we present the characterization of the bellows and our microblower-based pneumatic system as a whole, where each microblower is exclusively used to drive one bellows actuators. Firstly, we characterized the step-up response of the bellows by commanding a setpoint of 5 psi or 34.47 kPa, this has been empirically deemed the maximum pressure that the microblowers can sustain without requiring the microblowers to be at full output constantly. The results in Figure 6.10A show that the 20 mm bellows actuators have a rise time of 2.36 seconds, while the larger bellows have a rise time of 5.05 seconds. During step-down however, both bellows quickly deflate in roughly 1 second. A gentler ramp up from 0 to 34.47 kPa in 10 seconds and ramp down to 0 kPa again shown in Figure 6.10C, shows sufficient performance for use-cases that require slower responses. For reference, we included Figure 6.10D to show the frequency response of the 20 mm bellows with our system.

Pneumatic Band

With the goal of being able to vary the amount of pressure on the EMG electrodes in mind, we constructed a pneumatic band using our bellows and pneumatic system. A pair of dry EMG electrodes (forming one channel/site) was adhered to two of our bellows (one for each bellow) and the bellows were attached to a hook-and-loop strip. A large 50 mm wide bellows was attached to the end of the same hook-and-loop strip to form a pneumatic band that can simultaneously apply pressure on the electrodes and tighten the band on a forearm (Figure 6.1). We validate the performance of the pneumatic band by wrapping the band on a 3D printed jig with a force sensor (FMA series, Honeywell), as shown in Figure 6.10F, where one of the electrodes is placed directly on top of the force sensor. The jig has a circumference of approximately 208 mm. All three bellows were given a ramp up and ramp down signal up to 5 psi (34.47 kPa) as shown in Figure 6.10E. The results show that the band can achieve

a maximum quasi-static force of 4.1 N locally on the electrode. For comparison, strapping our EMG band on the characterization jig yielded a maximum force of approximately 4 N at the tightest setting. It should be noted that, according to [155] the bellows' force output is dependent on the amount of displacement, therefore, it is likely that the amount of force applied on a human forearm could differ from the the forces measured from the jig.

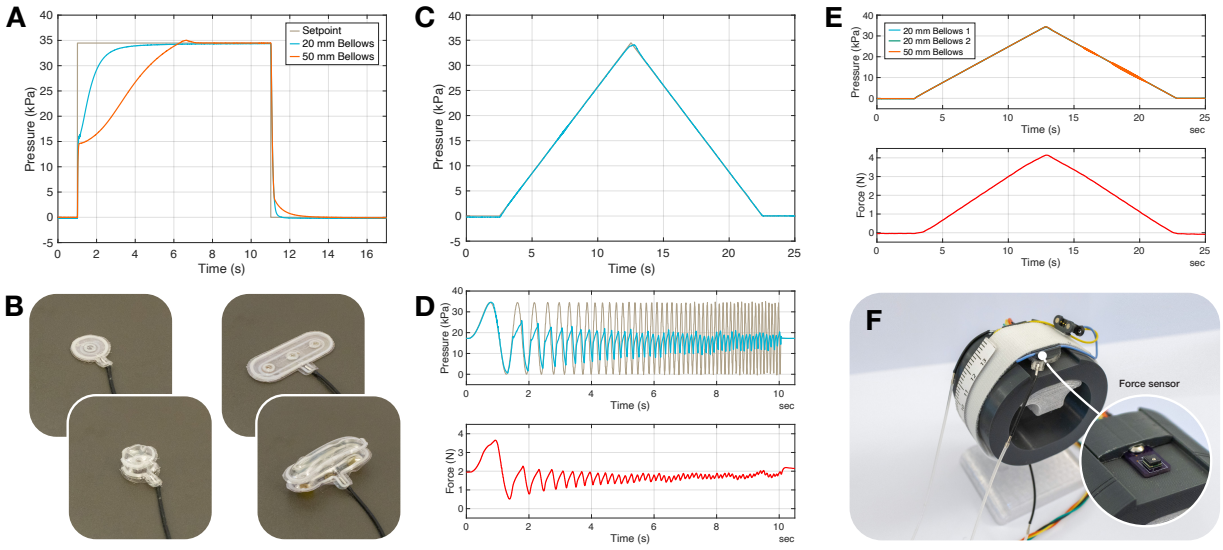

Figure 6.10: (A) The step response for both bellows have an initial instantaneous response, followed by a slower rise. (B) The bellows under characterization are shown in their deflated and inflated state. (C) A slower ramp response, akin to the actual usage of the bellows, shows good setpoint following. (D) The frequency response of the 20 mm bellows' and the resulting force of the frequency response test if all three bellows are installed on the band in (F). (E) The force response to a ramp input when all three bellows are installed on a band using the characterization jig in (F).

6.4 User Study and Data Collection

In order to better understand the effects of pressure on the skin-electrode interface and the resulting sEMG signals, we designed a user study based on hand-tracking applications in virtual reality with dual-intents: 1) to study the effects of varying conditions of skin-electrode coupling on the performance of using an EMG-based input modality, and 2) to collect sEMG data during said conditions for use in the proof-of-concept system for maintaining good coupling in Section 6.5. We chose hand-tracking as the main input modality as we hypothesize that EMG can serve as a supplementary modality to improve upon purely optical-based hand-tracking, as described in the following sections.

6.4.1 Experimental Setup

Hardware Setup

Our user study is run entirely on a Meta Quest 2 standalone virtual reality headset. The host microcontroller for our EMG band prototype and the pneumatic controller is connected to the headset via a USB hub. To minimize the amount of power line interference, a USB Power Delivery (USB-PD) power bank supplies power for the pneumatic controller. We connect three bellows on the pneumatic armband as described in Section 6.3.2 to the microblowers: two bellows modulate the amount of pressure on a pair of EMG electrodes (forming one channel) and a large bellows for increasing the overall pressure placed opposite of the electrodes on the arm (see Figure 6.7). Participants would don the EMG armband on their forearm as tightly as they feel comfortable and then the pneumatic armband would be worn adjacent to the armband, further up the arm, as shown in Figure 6.1. As we intend to emulate hardware that could be possibly used in everyday life, no preparation of the arm, such as shaving, abrasion or cleaning, is done prior to donning the armbands. The pair of electrodes on the pneumatic band is connected to the eighth channel of the EMG armband via a magnetic connector.

EMG as a Signal for Interaction Intention

The use of virtual reality in a productive or training environment can span many modes of interaction such as manipulating objects, browsing menus, or performing input through an interface. Here, we focus on two common interactions: the manipulation of three-dimensional physical objects in the virtual space and the interaction with a two-dimensional user interface using a cursor. Hand tracking provided natively by the headset is used as the main input modality, with our EMG system as a secondary input modality. In our study, the interaction with a physical object, such as turning a knob or pressing a button, is not permitted until the EMG activity reaches a certain amount defined by the moving average with a sliding window of 200 milliseconds of the absolute rectified value summed across all channels. Conversely, interaction with the two-dimensional interface is not permitted unless the EMG activity is lower than a predefined threshold. Thus, users would have to pinch or grab forcefully to interact with physical objects while two-dimensional screen-like user interfaces require minimal exertion—we define this as **EMG-modulated interaction**.

Consequently, as our system relies on the readings of a reliable sensor, having an electrode that is not properly contacting the skin can prevent users from activating their intended interaction. A loose electrode on one channel can introduce large signals from motion artifacts due to intermittent contact that overwhelms the rest of the channels, or lower the total amount of input due to a complete loss of contact. Here, we want to study the effect of having this condition of poor contact, therefore we use the pneumatic band to vary the amount of pressure applied on the electrodes. Low pressures, or none at all, should render the pneumatic band loose while high pressures should resemble the tightly worn EMG band.

Task and Virtual Environment

Participants are placed in an office-like virtual environment with a desk, monitor and a large two-dimensional panel 1 meter wide and 0.53 meters tall that emulates the native user interface found in virtual reality systems. The monitor is only used for diagnostics during the

study and displays the filtered real-time EMG signals from the EMG band rendered using ImPlot [2] and Dear ImGui [3]. Above the desk, 12 three-dimensional interactable objects are presented to the participant during most of the study. The objects are: a knob (similar to those found on stovetops), a slider, a large lever, an electrical plug and outlet, a dial, and a circular smart knob (like those found on smart thermostats) —further descriptions can be found in Figure 6.11. Every object has an on and off state, participants are to interact with each object to switch them into their on state as prompted, such as: the knob requires switching from pointing left to right, the circular smart knob must be rotated clockwise until the screen reads 100%, and the slider must be slid fully upwards. A small flag is raised above the objects to signal a success.

The two-dimensional panel (Figure 6.12) presents participants with three cursor-based tasks: 1) entering a four-digit keycode, 2) an icon and color matching task using two sliders, where one slider adjusts the icon and another adjusting the color, and 3) a drag and drop task to position the icon of a banana to its target outline. When prompted, participants would use a pointer and pinch-to-select style input to interact with the panel, where the pointer (or cursor) is positioned using a ray from the approximated location of the shoulder to the hand—which is also known as far-field targeting. We refer readers to the Oculus Developer[4] documentation for further reading on far-field targeting and pinch-to-select for hand tracking. All interactable components are driven using a modified version of the Oculus Interaction SDK[5] provided with the Oculus Integration package for Unity. We use the Bricklayer[6] package for Unity for running our user study.

---

[2] `https://github.com/epezent/implot`

[3] `https://github.com/ocornut/imgui`

[4] `https://developer.oculus.com/resources/hands-design-bp` - Last retrieved on June 25th, 2022

[5] `https://developer.oculus.com/documentation/unity/unity-isdk-interaction-sdk-overview` - Last retrieved on June 25th, 2022

[6] `https://github.com/prnthp/experiment-structures`

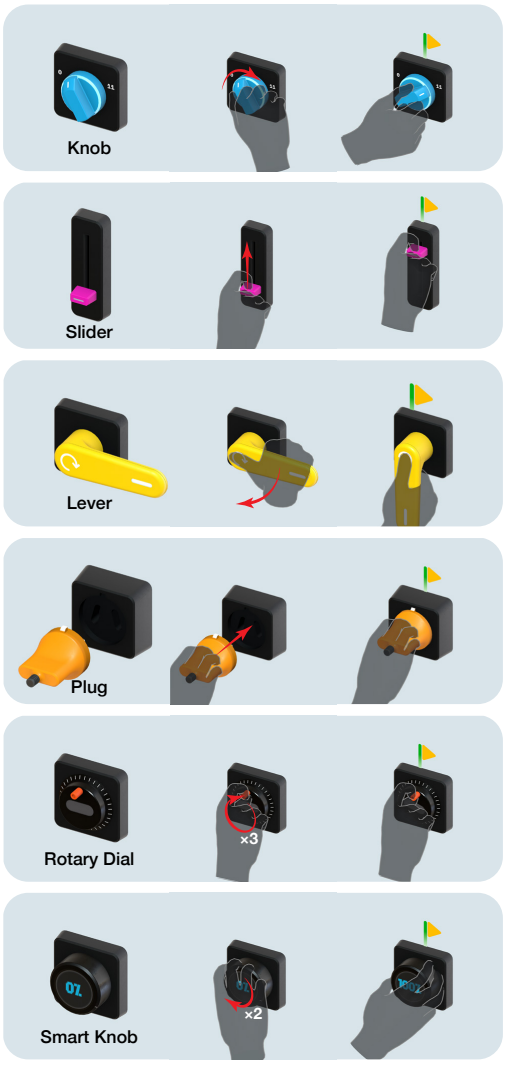

Figure 6.11: Physical object tasks. The *Knob* requires users to pinch and turn the knob clockwise by 50 degrees. The *Slider* requires users to slide the handle up by 10 cm. The *Lever* requires user to grab the handle and turn the lever 90 degrees in the direction shown. The *Plug* requires users to grab and push in the plug, which is constrained in the axis along the socket. The *Rotary Dial* requires users to pinch to grab the handle and spin the dial for three rotations clockwise. The *Smart Knob* requires users to grab onto and turn the knob's outer casing by approximately 180 degrees (the display will count up to 100%), which can be achieved by turning the knob 90 degrees twice.

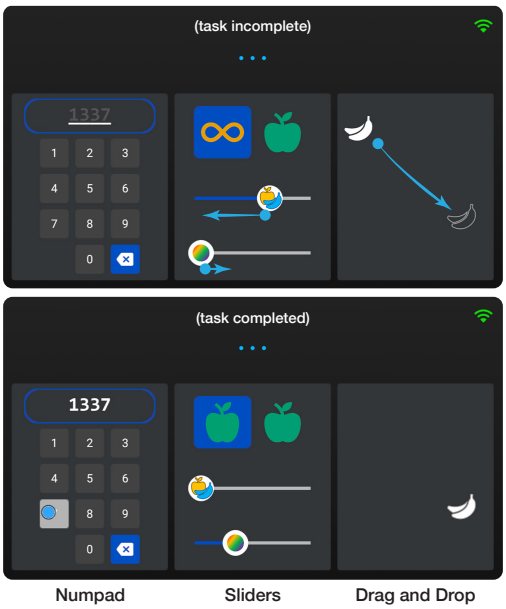

Figure 6.12: The three tasks on the 2D panel: (1) Number Pad. Users would enter a four digit code as shown in gray. If the user fails, the input restarts at the first digit. (2) Sliders. Users would slide the top slider to match the icon to the one on the left and the bottom slider to match the color. There are a total for 4 icons and 4 colors. (3) Drag and Drop. Users would hold down their cursor (pinch) to drag the banana icon to its outline location.

Study Flow

Our user study begins with participants first donning the EMG band tightly around the forearm followed by loosely wearing the pneumatic band adjacent to the EMG band nearer to the elbow with the pneumatic band's electrode adjacent to fourth channel of the EMG band (Figure 6.1). Participants then don the Meta Quest 2 headset and follow prompts shown on the 2D panel in the virtual environment. First, participants are given a short tutorial for each interactable object and cursor-based tasks, then, participants are given a short mock trial to acclimate to the task they would be doing: presented with 12 interactable physical objects and three cursor-based tasks, interact with each object and complete tasks as indicated by the green border one-by-one in rapid succession. As shown in Figure 6.13, the interactable objects are positioned directly in front of the user, with 6 objects in columns of 3 on the left and right, and the 2D panel is positioned approximately 1 meter away from the user. The interactable objects are intentionally positioned so the 2D panel is slightly occluded at each end—this is to ensure that there is overlap between the cursor's raycast and interactable objects, where users would have to use the EMG-modulated interaction to switch between modes of interaction. After the mock trial, participants then proceed to the actual trials. Each trial consists 40 consecutive interactions randomly switching between each of the 12 objects and 3 cursor-based tasks, with the objects re-positioned every 10 interactions. After each trial, a short 8 question questionnaire is presented on the 2D panel with questions relevant to the trial just completed.

A total of 10 trials are performed per session split into the following configurations: 2 trials where EMG-modulated interaction is disabled (therefore interactions can be engaged with simultaneously between objects and the cursor-based tasks), 2 trials where EMG-modulated interaction is enabled and the pneumatic band's electrodes are not used, 6 trials where the pneumatic band's electrodes are used (i.e. replacing channel 4 with channel 8) and the pressure of the of pneumatic band's bellows are set to either 0, 1, 2, 3, 4 or 5 pounds per square inch (psi). Note that in trials that the pneumatic band's electrodes are not used, the

Figure 6.13: A screenshot from within the virtual environment is shown. The physical objects in the foreground can be interacted with like their appearance suggests, such as a knob requiring a twist or a slider requiring a slide. The 2D panel resembles cursor-based or touch-based input similar to modern user interfaces found in VR, computers and smartphones. The monitor is used for diagnostics and is not interacted with during the user study.

bellows are set to 0 psi.

6.4.2 Results

We recruited 8 right-handed participants to participate in the study. The study was approved by our institution's Institutional Review Board (IRB) and all participants gave informed consent.

Task Performance

First, we evaluate the usability of our EMG-modulated interaction. Figure 6.15 shows that, on average, users required a higher overall time to complete both physical and panel-based tasks. Upon reviewing footage recorded from the headset during the study, we found that this increased time can be attributed to 1) users failing to generate enough muscle activity for physical tasks, thus having to retry the interaction and 2) users activating their forearm

muscles during panel-based tasks to aid in the initial aiming of the cursor thus having to relax their muscles in order to proceed. Nevertheless, our system does successfully prevent users from accidentally activating unintended tasks, as the overall amount of mistakes is lower when EMG-modulation was used (Figure 6.14, No Mod vs. Mod condition, p = 0.011).

When coupling pressure on the electrode was changed (Figure 6.14), users made more mistakes at lower pressures (0 to 2 psi), compared to higher pressure (3 to 5 psi) and the baseline (EMG). At lower coupling pressures, even if only a single electrode pair is contributing to the overall envelope, the resulting signal can be overwhelmed with motion artifacts, or lessen the overall signal, as presented in the next section. Motion artifacts overwhelming the signal causes the EMG-modulation to remain in the physical object interaction due to a large overall signal, while a lower than usual signal causes the cursor to engage with panel-based tasks.

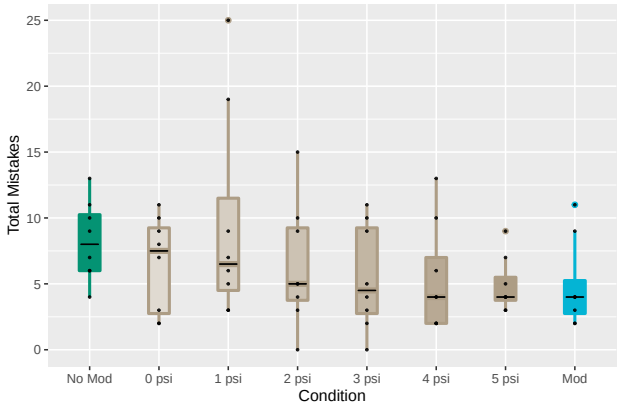

Figure 6.14: As the amount of pressure in the pneumatic band increases, the amount of engagements of the wrong object or task tends to decrease towards the baseline (Mod condition). The No Mod condition shown is when EMG-modulated interaction is not used.

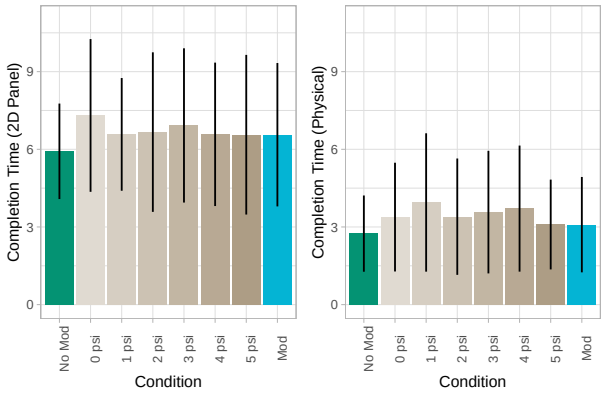

Figure 6.15: The completion time of tasks vary widely between each condition. The use of our EMG-modulated interaction increased the overall time for completion.

Effect of Coupling Pressure Variation on sEMG

The plots in Figure 6.16 are hand-picked representative EMG activity collected during the user study. We observed two behaviors when the pneumatic band is most loosened: 1) an overall decrease signal magnitude compared to the reference electrode (Figure 6.16A), 2) high intensity transients in signal magnitude that can be attributed to motion artifacts during tasks that require movement (Figure 6.16B). At higher pressures, when the pneumatic band is tighter against the arm, the EMG envelope between the pneumatic band's electrodes and reference electrodes closely resemble each other (Figure 6.16C & D).

Using the aforementioned observations, we analyzed the correlation (Pearson's correlation) between EMG envelopes between the pneumatic and reference electrode and the relative magnitude of signal compared to the reference electrode. Figure 6.17 shows a consistent lack of correlation between electrodes at 0 psi, and an increase in correlation as pressure increases. This increase, however, is not true for all subjects as some maintain a high correlation at 1 psi and above. Using the reference electrode as a baseline, when comparing the relative signal magnitude (calculated using the mean of the entire envelope during each trial), the signal magnitude has a trend of increasing along with pressure, possibly reaching a plateau

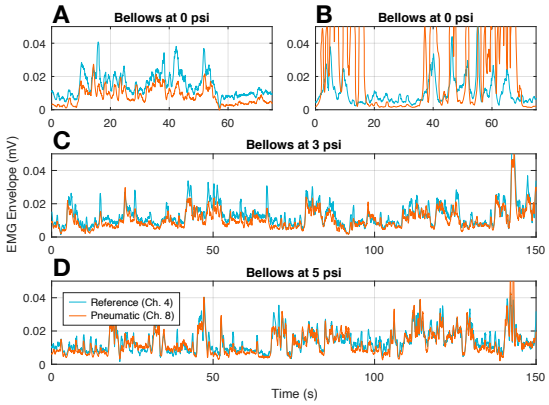

Figure 6.16: Representative examples of EMG activity at different amounts of pneumatic band pressure. (A) & (B) With the bellows deflated, the overall activity is noticeably reduced compared to the reference. Additionally, motion artifacts overwhelm the EMG readings in (B). (C) & (D) At 3 & 5 psi, the EMG activity at the pneumatic band is noticeably closer to that of the adjacent reference electrode. Note although the bands are adjacent, the muscle activity can never be exactly the same.

near 3 psi (Figure 6.17).

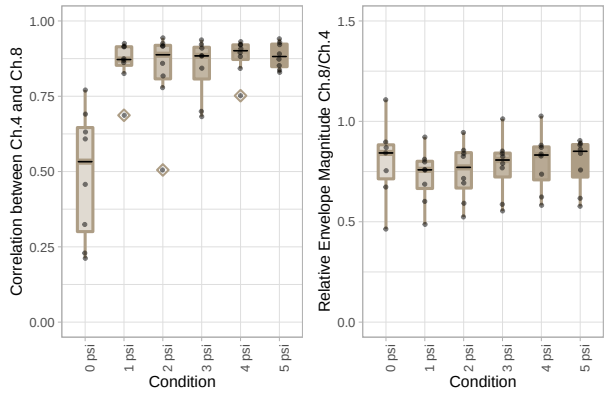

Figure 6.17: (Left) Correlation calculated between the EMG envelopes at varying levels of pressure reveal no correlation at 0 psi and high correlation at levels 1 psi and above. (Right) The relative overall EMG envelope magnitude when comparing the pneumatic band channel against the reference channel as a baseline.

Subjective Evaluation

After each trial, our participants answered a short questionnaire regarding their experience for the past set of tasks. Our users mostly reported that they did not perceive the pneumatic band as tight until the maximum amount of pressure at 5 psi. In terms of comfort, aside from a few instances, most subjects perceived the pneumatic band as comfortable (+2 level). In terms of difficulty, most subjects perceived the tasks as difficult at first, but reported less difficulty as the trials progressed, suggesting that there is a learning effect. There was no difference in difficulty between levels of pressure.

6.5 Proof-of-Concept Actuated Electrode System

In this section, we describe our efforts to build a closed-loop system for maintaining adequate contact between the electrode and the skin.

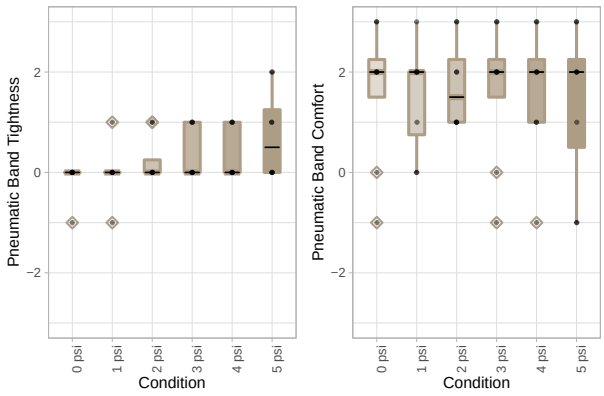

Figure 6.18: (Left) Participants reported that the pneumatic band was neither tight nor loose, aside from the highest pressure condition. (Right) Most participants reported that the band was comfortable at all pressure levels.

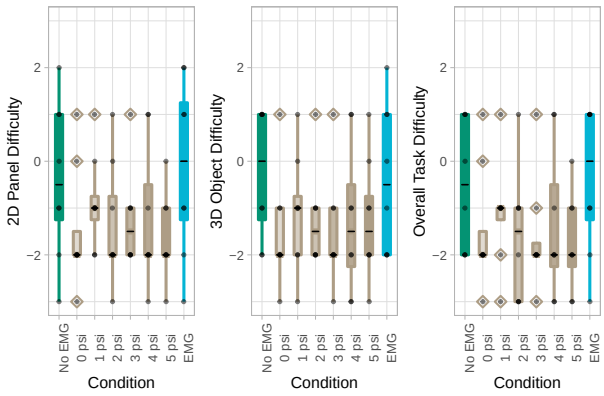

Figure 6.19: As the study progressed, participants reported less difficulty.

6.5.1 Dataset

As mentioned in Section 6.4, we also recorded the EMG readings during each trial. We use the readings to generate dataset that can be used for training or evaluation of a system that can determine the state of the electrode: good contact or poor contact. First, we segment each trial into individual interactions. Physical object interactions are segmented starting from the time the user's hand is hovering near the target object up until the object is successfully activated. Panel-based interactions are segmented starting from the time the user is prompted to complete the task to up until task completion. As our EMG-modulated interaction system relies on high and low muscle activity, we label physical object segments as high activity (`activity=1`) and panel-based segments as low activity (`activity=0`). As we are interested in the electrode contact state, we extract only the EMG channels of interest: channel 4 (reference) and channel 8 (pneumatic band). For each segment, we label their contact status according to the trial the segment is in. Segments from channel 4 are labeled as good contact (`contact=1`) regardless of the trial. Segments from channel 8 are labeled as poor contact (`contact=0`) in trials where the pneumatic band is at 0 psi. We chose to discard data from channel 8 where the pneumatic band is at 1 psi or higher due to the uncertainty of electrode state.

Feature Extraction

Our EMG data is sampled at 1000 Hz and contains a large amount of noisy, stochastic signals spanning a wide range of frequency. It has been shown that creating time-domain and frequency-domain features from windows of EMG data is beneficial for machine learning-based classifiers [101]. Therefore, we performed feature extraction to increase dimensionality of our EMG data. First, the raw EMG readings are high-pass filtered using a second-order Butterworth filter at 60 Hz, followed by notch filtering at 60 Hz and 120 Hz. Then, adapted from [82], we use the following features: **1)** mean absolute value, **2)** variance, **3)** log of the variance, **4)-6)** time-dependent power spectrum descriptors (TDPSD): sparseness,

irregularity factor and energy, **7)-11)** fast Fourier transform magnitudes at 15, 30, 60, 120 and 180 Hz. For each segment in our dataset, the data is split into 256 samples with a stride of 64 samples for feature extraction, which translates to approximately 256 milliseconds of data with a latency of 64 milliseconds for a real-time system. An excerpt of the resulting features is shown in Figure 6.20. In total our dataset consists of 355,366 samples with 11 features across all trials and subjects, with 243,439 samples labeled as good contact and 111,927 samples labeled as poor contact.

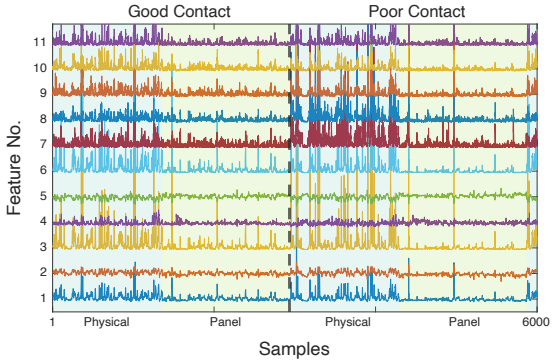

Figure 6.20: An excerpt of 6000 samples of our training dataset is shown. The features with skin-electrode contact labeled as poor contact (`contact=0`) show some distinct behavior from the those labeled as good contact (`contact=1`). The colored regions denote whether the samples are during physical tasks (`activity=1`) or panel tasks (`activity=0`).

6.5.2 State Estimation

Using the dataset's features as input and labels as state, we wish to estimate the state of electrode-skin contact. We constructed our neural network based on a Long short-term memory (LSTM) network. LSTM networks are thought to be well suited for timeseries prediction or classification and can utilize time-lagged dependencies [110]. Our network takes 5 samples (current + history of 4 previous samples) as input and classifies the state as either good or poor contact (binary classification). A diagram of the model is shown in

Figure 6.21A.

We constructed our machine learning pipeline using Keras [20]. To minimize class imbalance, random contiguous segments with good contact labels were dropped from the dataset until the number of samples in each class was approximately equal. Our neural network consists of a normalization layer, a fully connected layer of size 32 with ReLU activation, an LSTM layer with 24 units, and a fully connected output layer of size 1 with sigmoid activation. We deliberately built a small network to minimize computational time for deployment on a near real-time or embedded system. We trained our network using the Adam optimizer with a learning rate of 0.001 with decay, and binary crossentropy as the loss function. Our model is evaluated using binary accuracy as a metric, where a higher frequency of correct classification yields higher accuracy. After hyperparameter tuning, our state estimation model achieved an accuracy of 85% for our training dataset and 80% for our test dataset from an unseen subject (not contained in the training dataset).

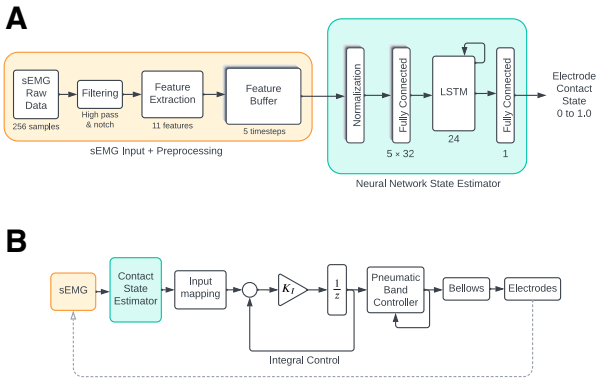

Figure 6.21: (A) A diagram feature extraction and neural network model based on an LSTM network is shown. The input to the model is a buffer of 5 timesteps (current + history of 4) of features extracted from 256 samples of filtered EMG. The model then classifies the contact state between good (1.0) and poor (0.0). (B) The control scheme for maintaining electrode contact is shown. Once the contact state is estimated, the integral controller builds up pressure in the pneumatic band if the contact state is poor.

6.5.3 Control

The resulting LSTM model was then saved as a TensorFlow Lite [26] model to be used in computing environments outside of Keras. Our model was then included into our Unity-based environment using the TensorFlow Lite for Unity Samples package [7]. A brief benchmark of the CPU execution time yielded 0.023 ms for LSTM state estimation and 0.015 ms for feature extraction averaged over 1000 subsequent calls on a Meta Quest 2 headset running a Unity-based Android application (IL2CPP + Burst compilation enabled) —which is well within the execution time required for real-time systems. Had there been longer execution times (in the order of tens of milliseconds), we would use a task-scheduler to delegate the computation to a separate thread, or the GPU (graphics processing unit with OpenCL) and DSP (digital signal processor via Arm Neon) for neural network execution and feature extraction, respectively.

Using the LSTM contact state estimation as input, we built an integral controller to adjust the amount of pressure in the pneumatic band. The controller has a sampling rate of 15 Hz to accommodate the slow response of the bellows and the sampling rate of our LSTM network. The integral controller behaves in such a way that pressure in the band is built up when the contact state is constantly poor, and maintains a relatively stable pressure when the electrode state transitions between good and poor.

6.5.4 Demonstration

We demonstrate a proof-of-concept system using the same task as our user study, presented a single trial. The trial is configured in the same manner as the EMG-modulated interaction condition in our user study, except our closed-loop system is turned on mid way through the trial. We collected the EMG data, our state estimator's predictions, and the pneumatic band's pressure. Upon observation of the collected data (Figure 6.22 it can be seen that in the duration that the control system is turned off, the state estimator correctly identifies

[7] `https://github.com/asus4/tf-lite-unity-sample`

events that are obvious motion artifacts. When the electrodes are stably resting on the skin (during tasks that involve the 2D panel), the contact state is reported to be good, hence, our state estimator generalizes favorably for detecting poor contact during motion, but does not generalize well for mostly idle situations. When the control system is turned on, once there is motion (physical object tasks), the state estimator detects the onset of a motion artifact and thus our band increases pressure. The remaining data of the excerpt from the trial shows that the EMG envelopes from the pneumatic band have consistently high correlation with the reference band when there is activity. The correlation between the EMG envelope of the pneumatic band's channel and the reference band's channel yielded 0.56 when the control system was off and 0.86 when the control system was on.

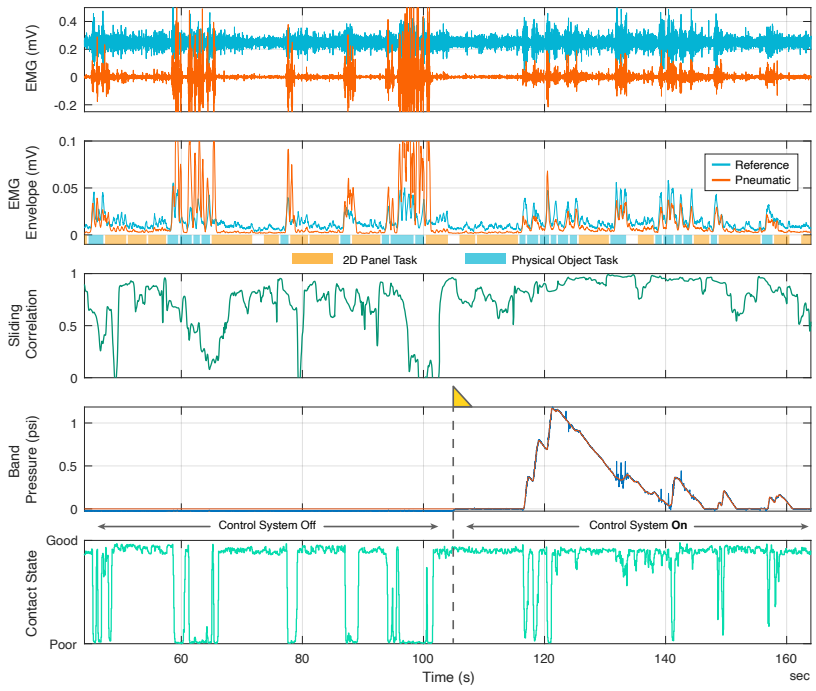

Figure 6.22: Excerpt of data collected during a modified trial from the user study. The flag (104 seconds) indicates the moment when the control system is turned on for the trial.

6.6 Limitations and Future Work

Our user study results indicate that for an EMG-based input modality, there is a need for active monitoring of the electrode-skin contact status. Using the linear envelope of EMG as part of an input modality in our study is fairly primitive compared to other techniques such as decoding hand activity or intent from lightly processed signals and features [131, 28], however, even our simple system suffers from an unreliable signal. We expect the issue to be exacerbated in systems that are more complex—especially those that utilize machine learning models or neural networks. Our results indicate that not much pressure is required to maintain good contact, however, there is still much variability due to the uniqueness of limb geometry and skin condition from person to person, despite our efforts of maintaining a similar initial tightness for the pneumatic band throughout our cohort. Our EMG dataset and contact estimation model are fairly unique to our particular setup, however, our device, prototypes, and the fabrication techniques presented here can be produced using consumer-grade tools such as 3D printers.

We believe that multiple facets of the work presented here can be incorporated into future research. Our EMG band provides a low-cost and lightweight alternative to research-grade devices and provides potential developers access to low-level interfaces (raw EMG signals, diagnostic commands) whilst maintaining plug-and-play capabilities with modern hardware. Our pneumatic system provides a platform for soft robotic devices that require low pneumatic volume without the need for bulky hardware such as pumps, valves and fittings. The system is also near solid-state, meaning the life expectancy is much higher than the electromechanical counterpart such as electric motor-based pumps. Our bellows fabrication technique can be further expanded to include more complex geometry or even integrated directly into larger 3D printed structures whilst printing on the same machine.

Chapter 7

CONCLUSIONS

Once a wearable device is worn on the body, the wearer's skin, underlying tissue, and sensory organs are mechanically and electrically coupled to the device, allowing the wearable device to sense and act upon the body. Mechanically, the dynamics of both systems can directly affect each other, propagating from the body of the device acting on the skin down to the sensory organs beneath the skin that react to vibration, pressure and temperature, allowing for haptic feedback. Electrically, the body generates minute electrical signals from neural activity and motor unit activation that can be picked up by the wearable device, allowing for these signals to be interpreted as input. A soft robotic system behaves similarly with respect to its environment. The compliance of soft robots naturally couple their dynamics strongly to the environment and often necessitate the use of equally compliant but difficult to interpret sensors. Thus, soft robotics share similar challenges to wearable devices and each domain can inform the design of the other. In this dissertation, we have presented a journey of soft robotic design from the fabrication of soft robots, how those designs allow complex sensing systems and how to gain interpretable readings from those complex sensors. Then, we elaborated on our application space of wearable devices by introducing two haptic systems that can behave differently depending on the mechanical coupling to the body. Lastly, we presented the application of soft robotic technologies to solve the overlooked issue of maintaining reliable contact with the body for a wearable input device.

The second chapter, **Negshell Casting** [108] introduced a novel method for combining hard and soft robotic elements together during fabrication to create more complex actuators and robotic systems. In the wearable device space, integrating rigid components into a soft, compliant structure is inevitable and Negshell Casting is one solution amongst many.

Soft robotic actuators seldom have rigid structures for linear sensing elements, furthermore, their behavior is also highly nonlinear with a large amount of noise and hysteresis as well. The third chapter, **Sensuator** [110] described a solution for state estimation using neural networks that thrive on nonlinearity. We presented a model that can learn to estimate the pose of a highly flexible soft robotic actuator using only its own actuation fluid as inputs. The wearable device equivalent can be the device itself, or even the human wearing the device, where we desire to estimate the state of the electrical or mechanical coupling of the device to the body. Haptic feedback through a wearable device is useless without knowing that the user can feel the feedback and likewise, sensing some biological signals from the body such as heart rate, electrocardiogram or electromyogram, can be unreliable if the device is not properly contacting the body.

In the fourth chapter, we introduced **Chasm** [106], a broadband haptic actuator, that laid the building block for studying how the human body perceives haptic feedback against different sites of mechanical grounding. We learned that rendering the same haptic effect on the head and fingertips generate a wholly different response both in terms of pure perceptual limits and the illusion afforded when rendering visual stimulus at the same time. Our fifth chapter, **Haplets** [109], took the concept of rendering haptic and visual stimulus in synchrony further and used the body itself to amplify haptic feedback. Haplets heavily utilized visual stimulus, hand tracking and the pseudo-haptic weight illusion, to create believable haptic feedback even if the actuator was not directly at the intended site. Haplets introduced a building block for wearable devices that are closely integrated with the human body.

Lastly, **Actually Active Electrodes**, our last chapter, introduced a building block for actively maintaining the aforementioned electromechanical grounding for wearable devices. We investigated how degraded inputs due to the poor mechanical grounding of input sensors on a wearable device (EMG) can effect the usability of a virtual reality application. Using soft robotic actuators, state estimation and control, we presented a solution by introducing a pneumatic band that can actively maintain adequate mechanical grounding for an EMG-based wearable input device.

We have set out to explore how soft robotics and wearable devices share a similar design space and can benefit from one another. Soft wearable devices are undoubtedly the ideal form of wearable devices for the human body —and some technologies are already available today. Yet, it is apparent that more work needs to be done to understand not only human-computer interaction, but human-computer integration as well, as we progress towards truly all-day wearable devices that seamlessly integrate with our everyday life.

BIBLIOGRAPHY

[1] Gizem Acar, Ozberk Ozturk, Ata Jedari Golparvar, Tamador Alkhidir Elboshra, Karl Böhringer, and Murat Kaya Yapici. Wearable and Flexible Textile Electrodes for Biopotential Signal Monitoring: A review. *Electronics*, 8(5):479, May 2019. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[2] Merwan Achibet, Benoît Le Gouis, Maud Marchal, Pierre-Alexandre Léziart, Ferran Argelaguet, Adrien Girard, Anatole Lécuyer, and Hiroyuki Kajimoto. FlexiFingers: Multi-finger interaction in VR combining passive haptics and pseudo-haptics. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 103–106, Los Angeles, CA, USA, March 2017. IEEE.

[3] Nathaniel Agharese, Tyler Cloyd, Laura H. Blumenschein, Michael Raitor, Elliot W. Hawkes, Heather Culbertson, and Allison M. Okamura. HapWRAP: Soft Growing Wearable Haptic Device. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5466–5472, South Brisbane, Australia, May 2018. IEEE. ISSN: 2577-087X.

[4] Petteri Aimonen. Nanopb - protocol buffers with small code size, December 2019.

[5] ams. AS5311 - High Resolution Linear Position Sensor – Hall IC - ams | ams, December 2019.

[6] Hideyuki Ando, Eisuke Kusachi, and Junji Watanabe. Nail-mounted tactile display for boundary/texture augmentation. In *Proceedings of the international conference on Advances in computer entertainment technology*, ACE '07, pages 292–293, New York, NY, USA, June 2007. Association for Computing Machinery.

[7] Apple. Detect Body and Hand Pose with Vision WWDC20, 2020.

[8] Héctor Barreiro, Joan Torres, and Miguel A. Otaduy. Natural Tactile Interaction with Virtual Clay. In *2021 IEEE World Haptics Conference (WHC)*, pages 403–408, Montreal, Canada, July 2021. IEEE.

[9] Hrvoje Benko, Christian Holz, Michael Sinclair, and Eyal Ofek. NormalTouch and TextureTouch: High-fidelity 3D Haptic Shape Rendering on Handheld Virtual Reality Controllers. In *UIST '16: Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 717–728, October 2016.

[10] Christopher Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer-Verlag, New York, 2006.

[11] David E. Caballero and Eric Rombokas. Sensitivity to Conflict between Visual Touch and Tactile Touch. *IEEE Transactions on Haptics*, 12(1):78–86, January 2019. Conference Name: IEEE Transactions on Haptics.

[12] Sonny Chan, Chase Tymms, and Nicholas Colonnese. Hasti: Haptic and Audio Synthesis for Texture Interactions. In *2021 IEEE World Haptics Conference (WHC)*, pages 733–738, July 2021.

[13] Karen B. Chen, Kevin Ponto, Ross D. Tredinnick, and Robert G. Radwin. Virtual Exertions: Evoking the Sense of Exerting Forces in Virtual Reality Using Gestures and Muscle Activity. *Human Factors*, 57(4):658–673, June 2015. Publisher: SAGE Publications Inc.

[14] Ke-Yu Chen, Shwetak N. Patel, and Sean Keller. Finexus: Tracking Precise Motions of Multiple Fingertips Using Magnetic Sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 1504–1514, New York, NY, USA, May 2016. Association for Computing Machinery.

[15] Nick Cheney, Robert MacCurdy, Jeff Clune, and Hod Lipson. Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. *ACM SIGEVOlution*, 7(1):11–23, August 2014.

[16] Keene Chin, Tess Hellebrekers, and Carmel Majidi. Machine Learning for Soft Robotic Sensing and Control. *Advanced Intelligent Systems*, 2(6):1900171, 2020.

[17] Inrak Choi, Heather Culbertson, Mark R. Miller, Alex Olwal, and Sean Follmer. Grabity: A Wearable Haptic Interface for Simulating Weight and Grasping in Virtual Reality. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, pages 119–130, New York, NY, USA, 2017. ACM. event-place: Québec City, QC, Canada.

[18] Inrak Choi, Evan Zhao, Eric J Gonzalez, and Sean Follmer. Augmenting Perceived Softness of Haptic Proxy Objects through Transient Vibration and Visuo-Haptic Illusion in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2020. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[19] Seungmoon Choi and Katherine J. Kuchenbecker. Vibrotactile Display: Perception, Technology, and Applications. *Proceedings of the IEEE*, 101(9):2093–2104, September 2013.

[20] François Chollet and Contributers. Keras, 2015.

[21] J. Chossat, Y. Park, R. J. Wood, and V. Duchaine. A Soft Strain Sensor Based on Ionic and Metal Liquids. *IEEE Sensors Journal*, 13(9):3405–3414, September 2013. Conference Name: IEEE Sensors Journal.

[22] Christian Cipriani, Marco D'Alonzo, and Maria Chiara Carrozza. A Miniature Vibrotactile Sensory Substitution Device for Multifingered Hand Prosthetics. *IEEE Transactions on Biomedical Engineering*, 59(2):400–408, February 2012. Conference Name: IEEE Transactions on Biomedical Engineering.

[23] Francesco Clemente and Christian Cipriani. A Novel Device for Multi-Modal Sensory Feedback in Hand Prosthetics: Design and Preliminary Prototype. In *2014 IEEE Haptics Symposium (HAPTICS)*, pages 569–573, February 2014. ISSN: 2324-7347.

[24] Salvador Cobos, Manuel Ferre, M.A. Sanchez Uran, Javier Ortego, and Cesar Pena. Efficient human hand kinematics for manipulation tasks. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2246–2251, September 2008. ISSN: 2153-0866.

[25] Cosimo Della Santina, Christian Duriez, and Daniela Rus. Model Based Control of Soft Robots: A Survey of the State of the Art and Open Challenges. *arXiv:2110.01358 [cs, eess]*, October 2021. arXiv: 2110.01358.

[26] TensorFlow Developers. TensorFlow, May 2022.

[27] Nathan Dunkelberger, Jenny Sullivan, Joshua Bradley, Nickolas P Walling, Indu Manickam, Gautam Dasarathy, Ali Israr, Frances W. Y. Lau, Keith Klumb, Brian Knott, Freddy Abnousi, Richard Baraniuk, and Marcia K O'Malley. Conveying Language Through Haptics: A Multi-sensory Approach. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, ISWC '18, pages 25–32, New York, NY, USA, 2018. ACM. event-place: Singapore, Singapore.

[28] Anany Dwivedi, Yongje Kwon, and Minas Liarokapis. EMG-Based Decoding of Manipulation Motions in Virtual Reality: Towards Immersive Interfaces. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3296–3303, Toronto, ON, Canada, October 2020. IEEE. ISSN: 2577-1655.

[29] Hesham Elsayed, Mayra Donaji Barrera Machuca, Christian Schaarschmidt, Karola Marky, Florian Müller, Jan Riemann, Andrii Matviienko, Martin Schmitz, Martin Weigel, and Max Mühlhäuser. VRSketchPen: Unconstrained Haptic Assistance for Sketching in Virtual 3D Environments. In *26th ACM Symposium on Virtual Reality*

Software and Technology, VRST '20, pages 1–11, New York, NY, USA, November 2020. Association for Computing Machinery.

[30] Cathy Mengying Fang and Chris Harrison. Retargeted Self-Haptics for Increased Immersion in VR without Instrumentation. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, UIST '21, pages 1109–1121, New York, NY, USA, October 2021. Association for Computing Machinery.

[31] FAULHABER. FAULHABER SR 1016 ... SR, December 2019.

[32] Ike Feitler. Direct fabrication of soft actuator with elastic resin.

[33] Jeremy A. Fishel and Gerald E. Loeb. Bayesian Exploration for Intelligent Identification of Textures. *Frontiers in Neurorobotics*, 6, June 2012.

[34] Formlabs. Cupping blowout.

[35] Andreas Frutiger, Joseph T. Muth, Daniel M. Vogt, Yiğit Mengüç, Alexandre Campo, Alexander D. Valentine, Conor J. Walsh, and Jennifer A. Lewis. Capacitive Soft Strain Sensors via Multicore–Shell Fiber Printing. *Advanced Materials*, 27(15):2440–2446, 2015.

[36] Adrien Girard, Maud Marchal, Florian Gosselin, Anthony Chabrier, François Louveau, and Anatole Lécuyer. HapTip: Displaying Haptic Shear Forces at the Fingertips for Multi-Finger Interaction in Virtual Environments. *Frontiers in Robotics and AI*, 2016.

[37] Brian Gleeson, Scott Horschel, and William Provancher. Design of a Fingertip-Mounted Tactile Display with Tangential Skin Displacement Feedback. *Haptics, IEEE Transactions on*, 3:297–301, January 2011.

[38] Google. MediaPipe Hands, 2020.

[39] Aakar Gupta, Majed Samad, Kenrick Kin, Per Ola Kristensson, and Hrvoje Benko. Investigating Remote Tactile Feedback for Mid-Air Text-Entry in Virtual Reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 350–360, November 2020. ISSN: 1554-7868.

[40] Sidhant Gupta, Tim Campbell, Jeffrey R. Hightower, and Shwetak N. Patel. SqueezeBlock: Using Virtual Springs in Mobile Devices for Eyes-free Interaction. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 101–104, New York, NY, USA, 2010. ACM. event-place: New York, New York, USA.

[41] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D. Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, Asaf Nitzan, Gang Dong, Yuting Ye, Lingling Tao, Chengde Wan, and Robert Wang. MEgATrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM Transactions on Graphics*, 39(4):87:1–87:13, July 2020.

[42] Shangchen Han, Beibei Liu, Robert Wang, Yuting Ye, Christopher D. Twigg, and Kenrick Kin. Online optical marker-based hand tracking with deep labels. *ACM Transactions on Graphics*, 37(4):1–10, August 2018.

[43] Faizan Haque, Mathieu Nancel, and Daniel Vogel. Myopoint: Pointing and Clicking Using Forearm Mounted Electromyography and Inertial Motion Sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3653–3656, New York, NY, USA, April 2015. Association for Computing Machinery.

[44] Tess Hellebrekers, Oliver Kroemer, and Carmel Majidi. Soft Magnetic Skin for Continuous Deformation Sensing. *Advanced Intelligent Systems*, 1(4):1900025, 2019.

[45] Tim Helps and Jonathan Rossiter. Proprioceptive Flexible Fluidic Actuators Using Conductive Working Fluids. *Soft Robotics*, 5(2):175–189, April 2018.

[46] Ronan Hinchet, Velko Vechev, Herbert Shea, and Otmar Hilliges. DextrES: Wearable Haptic Feedback for Grasping in VR via a Thin Form-Factor Electrostatic Brake. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, pages 901–912, New York, NY, USA, October 2018. Association for Computing Machinery.

[47] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-term Memory. *Neural computation*, 9:1735–80, December 1997.

[48] Meng-Ju Hsieh, Rong-Hao Liang, and Bing-Yu Chen. NailTactors: eyes-free spatial output using a nail-mounted tactor array. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '16, pages 29–34, New York, NY, USA, September 2016. Association for Computing Machinery.

[49] Phillip Hyatt, David Wingate, and Marc D. Killpack. Model-Based Control of Soft Actuators Using Learned Non-linear Discrete-Time Models. *Frontiers in Robotics and AI*, 6, 2019. Publisher: Frontiers.

[50] Texas Instruments. ADS1299 Low-Noise, 4-, 6-, 8-Channel, 24-Bit, Analog-to-Digital Converter for EEG and Biopotential Measurements, January 2017.

[51] Texas Instruments. DRV8835 1.5A low voltage stepper or single/dual brushed DC motor driver w/ dual supplies (PWM or PH/EN ctrl) | TI.com, December 2019.

[52] Alexandra Ion, Edward Jay Wang, and Patrick Baudisch. Skin Drag Displays: Dragging a Physical Tactor Across the User's Skin Produces a Stronger Tactile Stimulus Than Vibrotactile. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 2501–2504, New York, NY, USA, 2015. ACM. event-place: Seoul, Republic of Korea.

[53] Ali Israr, Siyan Zhao, and Oliver Schneider. Exploring Embedded Haptics for Social Networking and Interactions. In *CHI EA '15: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 1899–1904, April 2015.

[54] Ali Israr, Siyan Zhao, Zachary Schwemler, and Adam Fritz. Stereohaptics Toolkit for Dynamic Tactile Experiences. In Constantine Stephanidis, editor, *HCI International 2019 – Late Breaking Papers*, Lecture Notes in Computer Science, pages 217–232, Cham, 2019. Springer International Publishing.

[55] Andrés Jaramillo-Yánez, Marco E. Benalcázar, and Elisa Mena-Maldonado. Real-Time Hand Gesture Recognition Using Surface Electromyography and Machine Learning: A Systematic Literature Review. *Sensors*, 20(9):2467, January 2020. Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.

[56] Roland S. Johansson and J. Randall Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 10(5):345–359, May 2009. Number: 5 Publisher: Nature Publishing Group.

[57] Kenneth O. Johnson. The Roles and Functions of Cutaneous Mechanoreceptors. *Current Opinion in Neurobiology*, 11(4):455–461, August 2001.

[58] Laura Kalevo, Tomi Miettinen, Akseli Leino, Samu Kainulainen, Henri Korkalainen, Katja Myllymaa, Juha Töyräs, Timo Leppänen, Tiina Laitinen, and Sami Myllymaa. Effect of Sweating on Electrode-Skin Contact Impedances and Artifacts in EEG Recordings With Various Screen-Printed Ag/Agcl Electrodes. *IEEE Access*, 8:50934–50943, 2020. Conference Name: IEEE Access.

[59] Tom Kalisky, Yueqi Wang, Benjamin Shih, Dylan Drotman, Saurabh Jadhav, Eliah Aronoff-Spencer, and Michael Tolley. Differential pressure control of 3D printed soft

fluidic actuators. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6207–6213, September 2017.

[60] Jakob Karolus, Annika Kilian, Thomas Kosch, Albrecht Schmidt, and Paweł W. Wozniak. Hit the Thumb Jack! Using Electromyography to Augment the Piano Keyboard. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, pages 429–440. Association for Computing Machinery, New York, NY, USA, July 2020.

[61] Robert K. Katzschmann, Andrew D. Marchese, and Daniela Rus. Hydraulic Autonomous Soft Robotic Fish for 3D Swimming. In M. Ani Hsieh, Oussama Khatib, and Vijay Kumar, editors, *Experimental Robotics: The 14th International Symposium on Experimental Robotics*, Springer Tracts in Advanced Robotics, pages 405–420. Springer International Publishing, Cham, 2016.

[62] E. J. Kelly, G. Terenghi, A. Hazari, and M. Wiberg. Nerve fibre and sensory end organ density in the epidermis and papillary dermis of the human hand. *British Journal of Plastic Surgery*, 58(6):774–779, September 2005.

[63] Haydon Kerk. Kerk 5/64" (2mm) Lead Screw, December 2019.

[64] Wolf Kienzle, Eric Whitmire, Chris Rittaler, and Hrvoje Benko. ElectroRing: Subtle Pinch and Touch Detection with a Ring. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–12, New York, NY, USA, May 2021. Association for Computing Machinery.

[65] Dongeon Kim, Jonghak Lee, Wan-Young Chung, and Jangmyung Lee. Artificial Intelligence-Based Optimal Grasping Control. *Sensors*, 20(21):6390, January 2020. Number: 21 Publisher: Multidisciplinary Digital Publishing Institute.

[66] Hwan Kim, HyeonBeom Yi, Hyein Lee, and Woohun Lee. HapCube: A Wearable Tactile Device to Provide Tangential and Normal Pseudo-Force Feedback on a Fingertip. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 1–13, New York, NY, USA, April 2018. Association for Computing Machinery.

[67] Lawrence H. Kim, Pablo Castillo, Sean Follmer, and Ali Israr. VPS Tactile Display: Tactile Information Transfer of Vibration, Pressure, and Shear. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(2):51:1–51:17, June 2019.

[68] Mingyu Kim, Jinmo Kim, Kisung Jeong, and Changhun Kim. Grasping VR: Presence of Pseudo-Haptic Interface Based Portable Hand Grip System

in Immersive Virtual Reality. *International Journal of Human–Computer Interaction*, 36(7):685–698, April 2020. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10447318.2019.1680920.

[69] Anastasia Koivikko and Veikko Sariola. Fabrication of Soft Devices with Buried Fluid Channels by Using Sacrificial 3D Printed Molds. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 509–513, April 2019.

[70] Peter Konrad. *The ABC of EMG: A Practical Introduction to Kinesiological Electromyography*, volume 1. Noraxon U.S.A., Scottsdale, Arizona, USA, 2005.

[71] K.J. Kuchenbecker, J. Fiene, and G. Niemeyer. Improving contact realism through event-based haptic feedback. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):219–230, March 2006. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[72] Meta Reality Labs. Inside Facebook Reality Labs: Wrist-based interaction for the next computing platform, March 2021.

[73] Anatole Lecuyer, Sabine Coquillart, Abderrahmane Kheddar, Paul Richard, and Philippe Coiffet. Pseudo-Haptic Feedback: Can Isometric Input Devices Simulate Force Feedback? In *Proceedings of the IEEE Virtual Reality 2000 Conference*, page 83. IEEE Computer Society, March 2000.

[74] Jaeyeon Lee, Mike Sinclair, Mar Gonzalez-Franco, Eyal Ofek, and Christian Holz. TORC: A Virtual Reality Controller for In-Hand High-Dexterity Finger Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13. Association for Computing Machinery, New York, NY, USA, May 2019.

[75] Gábor Lengyel, Goda Žalalytė, Alexandros Pantelides, James N Ingram, József Fiser, Máté Lengyel, and Daniel M Wolpert. Unimodal statistical learning produces multimodal object-like representations. *eLife*, 8:e43942, May 2019. Publisher: eLife Sciences Publications, Ltd.

[76] C. Li, P. Wu, S. Lee, A. Gorton, M. J. Schulz, and C. H. Ahn. Flexible Dome and Bump Shape Piezoelectric Tactile Sensors Using PVDF-TrFE Copolymer. *Journal of Microelectromechanical Systems*, 17(2):334–341, April 2008. Conference Name: Journal of Microelectromechanical Systems.

[77] Yilin Liu, Shijia Zhang, and Mahanth Gowda. NeuroPose: 3D Hand Pose Tracking using EMG Wearables. In *Proceedings of the Web Conference 2021*, WWW '21, pages 1471–1482, New York, NY, USA, April 2021. Association for Computing Machinery.

[78] Jasmine Lu, Ziwei Liu, Jas Brooks, and Pedro Lopes. Chemical Haptics: Rendering Haptic Sensations via Topical Stimulants. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 239–257, Virtual Event USA, October 2021. ACM.

[79] Anatole Lécuyer. Simulating Haptic Feedback Using Vision: A Survey of Research and Applications of Pseudo-Haptic Feedback. *Presence*, 18(1):39–53, February 2009. Conference Name: Presence.

[80] Raymond R. Ma, Lael U. Odhner, and Aaron M. Dollar. A modular, open-source 3D printed underactuated hand. In *2013 IEEE International Conference on Robotics and Automation*, pages 2737–2743, Karlsruhe, Germany, May 2013. IEEE.

[81] Jens Maiero, David Eibich, Ernst Kruijff, Andre Hinkenjann, Wolfgang Stuerzlinger, Hrvoje Benko, and Gheorghita Ghinea. Back-of-Device Force Feedback Improves Touchscreen Interaction for Mobile Devices. *IEEE transactions on haptics*, 12(4):483–496, December 2019.

[82] Itzel Jared Rodríguez Martínez, Andrea Mannini, Francesco Clemente, and Christian Cipriani. Online Grasp Force Estimation From the Transient EMG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(10):2333–2341, October 2020. Conference Name: IEEE Transactions on Neural Systems and Rehabilitation Engineering.

[83] D. Ian McCloskey. Kinesthesia, Kinesthetic Perception. In Jeremy M. Wolfe, editor, *Sensory Systems: II: Senses Other than Vision*, Readings from the Encyclopedia of Neuroscience, pages 36–38. Birkhäuser, Boston, MA, 1988.

[84] Edward F. Melcer, Michael T. Astolfi, Mason Remaley, Adam Berenzweig, and Tudor Giurgica-Tiron. CTRL-Labs: Hand Activity Estimation and Real-time Control from Neuromuscular Signals. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI EA '18, pages 1–4, New York, NY, USA, April 2018. Association for Computing Machinery.

[85] Y. Mengüç, Y. Park, E. Martinez-Villalpando, P. Aubin, M. Zisook, L. Stirling, R. J. Wood, and C. J. Walsh. Soft wearable motion sensing suit for lower limb biomechanics measurements. In *2013 IEEE International Conference on Robotics and Automation*, pages 5309–5316, May 2013. ISSN: 1050-4729.

[86] Roberto Merletti and Dario Farina, editors. *Surface Electromyography: Physiology, Engineering, and Applications*. Wiley-IEEE Press, Piscataway,NJ, USA, 1st edition edition, April 2016.

[87] Kouta Minamizawa, Souichiro Fukamachi, Hiroyuki Kajimoto, Naoki Kawakami, and Susumu Tachi. Gravity Grabber: Wearable Haptic Display to Present Virtual Mass Sensation. In *ACM SIGGRAPH 2007 Emerging Technologies*, SIGGRAPH '07, New York, NY, USA, 2007. ACM. event-place: San Diego, California.

[88] Kouta Minamizawa, Domenico Prattichizzo, and Susumu Tachi. Simplified Design of Haptic Display by Extending One-Point Kinesthetic Feedback to Multipoint Tactile Feedback. In *2010 IEEE Haptics Symposium*, pages 257–260, March 2010. ISSN: 2324-7355.

[89] Stefano Mintchev, Marco Salerno, Alexandre Cherpillod, Simone Scaduto, and Jamie Paik. A Portable Three-Degrees-of-Freedom Force Feedback Origami Robot for Human–Robot Interactions. *Nature Machine Intelligence*, 1(12):584–593, December 2019.

[90] Bobak Mosadegh, Panagiotis Polygerinos, Christoph Keplinger, Sophia Wennstedt, Robert F. Shepherd, Unmukt Gupta, Jongmin Shim, Katia Bertoldi, Conor J. Walsh, and George M. Whitesides. Pneumatic Networks for Soft Robotics that Actuate Rapidly. *Advanced Functional Materials*, 24(15):2163–2170, 2014.

[91] Florian Floyd Mueller, Pedro Lopes, Paul Strohmeier, Wendy Ju, Caitlyn Seim, Martin Weigel, Suranga Nanayakkara, Marianna Obrist, Zhuying Li, Joseph Delfa, Jun Nishida, Elizabeth M. Gerber, Dag Svanaes, Jonathan Grudin, Stefan Greuter, Kai Kunze, Thomas Erickson, Steven Greenspan, Masahiko Inami, Joe Marshall, Harald Reiterer, Katrin Wolf, Jochen Meyer, Thecla Schiphorst, Dakuo Wang, and Pattie Maes. Next Steps for Human-Computer Integration. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–15. Association for Computing Machinery, New York, NY, USA, April 2020.

[92] Kohei Nakajima, Helmut Hauser, Tao Li, and Rolf Pfeifer. Exploiting the Dynamics of Soft Materials for Machine Learning. *Soft Robotics*, 5(3):339–347, April 2018. Publisher: Mary Ann Liebert, Inc., publishers.

[93] Xin Niu, Xinhua Gao, Yuefeng Liu, and Hao Liu. Surface bioelectric dry Electrodes: A review. *Measurement*, 183:109774, October 2021.

[94] Claudio Pacchierotti, Gionata Salvietti, Irfan Hussain, Leonardo Meli, and Domenico Prattichizzo. The hRing: A Wearable Haptic Device to Avoid Occlusions in Hand Tracking. In *2016 IEEE Haptics Symposium (HAPTICS)*, pages 134–139, April 2016. ISSN: 2324-7355.

[95] Min Pan, Chenggang Yuan, Hastha Anpalagan, Andrew Plummer, Jun Zou, Junhui Zhang, and Chris Bowen. Soft Controllable Carbon Fibre-based Piezoresistive

Self-Sensing Actuators. *Actuators*, 9(3):79, September 2020. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[96] Farshid Salemi Parizi, Eric Whitmire, and Shwetak Patel. AuraRing: Precise Electromagnetic Finger Tracking. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(4):150:1–150:28, December 2019.

[97] Dr. Dinesh Patel, Amir Hosein Sakhaei, Michael Layani, Biao Zhang, Qi Ge, and Shlomo Magdassi. Highly Stretchable and UV Curable Elastomers for Digital Light Processing Based 3D Printing. *Advanced Materials*, 29:1606000, February 2017.

[98] Yi-Hao Peng, Carolyn Yu, Shi-Hong Liu, Chung-Wei Wang, Paul Taele, Neng-Hao Yu, and Mike Y. Chen. WalkingVibe: Reducing Virtual Reality Sickness and Improving Realism while Walking in VR using Unobtrusive Head-mounted Vibrotactile Feedback. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–12, New York, NY, USA, April 2020. Association for Computing Machinery.

[99] Evan Pezent, Brandon Cambio, and Marcia K. O'Malley. Syntacts: Open-Source Software and Hardware for Audio-Controlled Haptics. *IEEE Transactions on Haptics*, 14(1):225–233, January 2021. Conference Name: IEEE Transactions on Haptics.

[100] Evan Pezent, Ali Israr, Majed Samad, Shea Robinson, Priyanshu Agarwal, Hrvoje Benko, and Nick Colonnese. Tasbi: Multisensory Squeeze and Vibrotactile Wrist Haptics for Augmented and Virtual Reality. In *2019 IEEE World Haptics Conference (WHC)*, pages 1–6, July 2019.

[101] Angkoon Phinyomark, Rami N. Khushaba, and Erik Scheme. Feature Extraction and Selection for Myoelectric Control Based on Wearable EMG Sensors. *Sensors*, 18(5):1615, May 2018. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[102] Sirawat Pitaksarit. 5argon/protobuf-unity, December 2019. original-date: 2017-07-06T14:48:05Z.

[103] Christopher J. Ploch, Jung Hwa Bae, Wendy Ju, and Mark Cutkosky. Haptic Skin Stretch on A Steering Wheel for Displaying Preview Information in Autonomous Cars. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 60–65, October 2016. ISSN: 2153-0866.

[104] Ivan Poupyrev and Shigeaki Maruyama. Tactile Interfaces for Small Touch Screens. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and*

Technology, UIST '03, pages 217–220, New York, NY, USA, 2003. ACM. event-place: Vancouver, Canada.

[105] Domenico Prattichizzo, Francesco Chinello, Claudio Pacchierotti, and Monica Malvezzi. Towards Wearability in Fingertip Haptics: A 3-DoF Wearable Device for Cutaneous Force Feedback. *Haptics, IEEE Transactions on*, 6:506–516, October 2013.

[106] Pornthep Preechayasomboon, Ali Israr, and Majed Samad. Chasm: A Screw Based Expressive Compact Haptic Actuator. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–13, New York, NY, USA, April 2020. Association for Computing Machinery.

[107] Pornthep Preechayasomboon and Eric Rombokas. ConTact Sensors: A Tactile Sensor Readily Integrable into Soft Robots. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 605–610, April 2019.

[108] Pornthep Preechayasomboon and Eric Rombokas. Negshell casting: 3D-printed structured and sacrificial cores for soft robot fabrication. *PLOS ONE*, 15(6):e0234354, June 2020. Publisher: Public Library of Science.

[109] Pornthep Preechayasomboon and Eric Rombokas. Haplets: Finger-Worn Wireless and Low-Encumbrance Vibrotactile Haptic Feedback for Virtual and Augmented Reality. *Frontiers in Virtual Reality*, 2:121, 2021.

[110] Pornthep Preechayasomboon and Eric Rombokas. Sensuator: A Hybrid Sensor–Actuator Approach to Soft Robotic Proprioception Using Recurrent Neural Networks. *Actuators*, 10(2):30, February 2021. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[111] Nicolaas Prins and Frederick A. A. Kingdom. Applying the Model-Comparison Approach to Test Specific Research Hypotheses in Psychophysical Research Using the Palamedes Toolbox. *Frontiers in Psychology*, 9, July 2018.

[112] William R. Provancher, Mark R. Cutkosky, Katherine J. Kuchenbecker, and Günter Niemeyer. Contact Location Display for Haptic Perception of Curvature and Object Motion. *International Journal of Robotics Research*, 24(9):691–702, September 2005.

[113] William R. Provancher and Nicholas D. Sylvester. Fingerpad Skin Stretch Increases the Perception of Virtual Friction. *IEEE Transactions on Haptics*, 2(4):212–223, October 2009.

[114] Zhan Fan Quek, Samuel B. Schorr, Ilana Nisky, William R. Provancher, and Allison M. Okamura. Sensory Substitution and Augmentation Using 3-Degree-of-Freedom Skin Deformation Feedback. *IEEE Transactions on Haptics*, 8(2):209–221, April 2015.

[115] Lorcan Reidy, Dennis Chan, Charles Nduka, and Hatice Gunes. Facial Electromyography-based Adaptive Virtual Reality Gaming for Cognitive Training. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, pages 174–183. Association for Computing Machinery, New York, NY, USA, October 2020.

[116] Jun Rekimoto. SenseableRays: opto-haptic substitution for touch-enhanced interactive spaces. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems - CHI EA '09*, page 2519, Boston, MA, USA, 2009. ACM Press.

[117] Michael Rietzler, Gabriel Haas, Thomas Dreja, Florian Geiselhart, and Enrico Rukzio. Virtual Muscle Force: Communicating Kinesthetic Forces Through Pseudo-Haptic Feedback and Muscle Input. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, pages 913–922, New York, NY, USA, October 2019. Association for Computing Machinery.

[118] Nataliya Rokhmanova and Eric Rombokas. Vibrotactile Feedback Improves Foot Placement Perception on Stairs for Lower-Limb Prosthesis Users. In *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, pages 1215–1220, June 2019. ISSN: 1945-7901.

[119] Steeven Villa Salazar, Claudio Pacchierotti, Xavier de Tinguy, Anderson Maciel, and Maud Marchal. Altering the Stiffness, Friction, and Shape Perception of Tangible Objects in Virtual Reality Using Wearable Haptics. *IEEE Transactions on Haptics*, 13(1):167–174, January 2020. Conference Name: IEEE Transactions on Haptics.

[120] Majed Samad, Elia Gatti, Anne Hermes, Hrvoje Benko, and Cesare Parise. Pseudo-Haptic Weight: Changing the Perceived Weight of Virtual Objects By Manipulating Control-Display Ratio. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, Glasgow Scotland Uk, May 2019. ACM.

[121] C. D. Santina, R. L. Truby, and D. Rus. Data–Driven Disturbance Observers for Estimating External Forces on Soft Robots. *IEEE Robotics and Automation Letters*, 5(4):5717–5724, October 2020. Conference Name: IEEE Robotics and Automation Letters.

[122] Samuel B. Schorr and Allison M. Okamura. Fingertip Tactile Devices for Virtual Object Manipulation and Exploration. In *Proceedings of the 2017 CHI Conference on Human*

Factors in Computing Systems, CHI '17, pages 3115–3119, New York, NY, USA, May 2017. Association for Computing Machinery.

[123] L. Seminara, L. Pinna, M. Valle, L. Basiricò, A. Loi, P. Cosseddu, A. Bonfiglio, A. Ascia, M. Biso, A. Ansaldo, D. Ricci, and G. Metta. Piezoelectric Polymer Transducer Arrays for Flexible Tactile Sensors. *IEEE Sensors Journal*, 13(10):4022–4029, October 2013. Conference Name: IEEE Sensors Journal.

[124] Suranga Seneviratne, Yining Hu, Tham Nguyen, Guohao Lan, Sara Khalifa, Kanchana Thilakarathna, Mahbub Hassan, and Aruna Seneviratne. A Survey of Wearable Devices and Challenges. *IEEE Communications Surveys Tutorials*, 19(4):2573–2620, 2017. Conference Name: IEEE Communications Surveys Tutorials.

[125] Yu She, Chang Li, Jonathon Cleary, and Hai-Jun Su. Design and Fabrication of a Soft Robotic Hand With Embedded Actuators and Sensors. *Journal of Mechanisms and Robotics*, 7(2), May 2015.

[126] Yu She, Sandra Q. Liu, Peiyu Yu, and Edward Adelson. Exoskeleton-covered soft finger with vision-based proprioception and tactile sensing. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10075–10081, May 2020. ISSN: 2577-087X.

[127] Jotaro Shigeyama, Takeru Hashimoto, Shigeo Yoshida, Takuji Narumi, Tomohiro Tanikawa, and Michitaka Hirose. Transcalibur: A Weight Shifting Virtual Reality Controller for 2D Shape Rendering based on Computational Perception Model. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–11. Association for Computing Machinery, New York, NY, USA, May 2019.

[128] Ali Shtarbanov. FlowIO Development Platform – the Pneumatic "Raspberry Pi" for Soft Robotics. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–6, Yokohama Japan, May 2021. ACM.

[129] Astrini Sie, David Boe, and Eric Rombokas. Design and Evaluation of a Wearable Haptic Feedback System for Lower Limb Prostheses During Stair Descent. In *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, pages 219–224, August 2018. ISSN: 2155-1782.

[130] Miguel Simão, Nuno Mendes, Olivier Gibaru, and Pedro Neto. A Review on Electromyography Decoding and Pattern Recognition for Human-Machine Interaction. *IEEE Access*, 7:39564–39582, 2019. Conference Name: IEEE Access.

[131] Miguel Simão, Pedro Neto, and Olivier Gibaru. EMG-based online classification of gestures with recurrent neural networks. *Pattern Recognition Letters*, 128:45–51, December 2019.

[132] Pratheev Sreetharan, Ali Israr, and Priyanshu Agarwal. A Compact Skin-Shear Device using a Lead-Screw Mechanism. In *2019 IEEE World Haptics Conference (WHC)*, pages 527–532, July 2019.

[133] C. B. Teeple, K. P. Becker, and R. J. Wood. Soft Curvature and Contact Force Sensors for Deep-Sea Grasping via Soft Optical Waveguides. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1621–1627, October 2018. ISSN: 2153-0866.

[134] Shan-Yuan Teng, Pengyu Li, Romain Nith, Joshua Fonseca, and Pedro Lopes. Touch&Fold: A Foldable Haptic Actuator for Rendering Touch in Mixed Reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–14, Yokohama Japan, May 2021. ACM.

[135] Carly Thalman and Panagiotis Artemiadis. A review of soft wearable robots that provide active assistance: Trends, common actuation methods, fabrication, and applications. *Wearable Technologies*, 1, 2020. Publisher: Cambridge University Press.

[136] Carl J. Thrasher, Johanna J. Schwartz, and Andrew J. Boydston. Modular Elastomer Photoresins for Digital Light Processing Additive Manufacturing. *ACS Applied Materials & Interfaces*, 9(45):39708–39716, November 2017.

[137] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi. Model-Based Reinforcement Learning for Closed-Loop Dynamic Control of Soft Robotic Manipulators. *IEEE Transactions on Robotics*, 35(1):124–134, February 2019. Conference Name: IEEE Transactions on Robotics.

[138] 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), January 2019. Publisher: Science Robotics Section: Research Article.

[139] Lei Tian, Aiguo Song, and Dapeng Chen. A Novel Haptic Stylus for Mobile Terminal. In Fernando Bello, Hiroyuki Kajimoto, and Yon Visell, editors, *Haptics: Perception, Devices, Control, and Applications*, Lecture Notes in Computer Science, pages 338–349, Cham, 2016. Springer International Publishing.

[140] C. To, T. L. Hellebrekers, and Y. Park. Highly stretchable optical sensors for pressure, strain, and curvature measurement. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5898–5903, September 2015.

[141] R. L. Truby, C. D. Santina, and D. Rus. Distributed Proprioception of 3D Configuration in Soft, Sensorized Robots via Deep Learning. *IEEE Robotics and Automation Letters*, 5(2):3299–3306, April 2020. Conference Name: IEEE Robotics and Automation Letters.

[142] 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, April 2019.

[143] Ryan L. Truby, Michael Wehner, Abigail K. Grosskopf, Daniel M. Vogt, Sebastien G. M. Uzel, Robert J. Wood, and Jennifer A. Lewis. Soft Somatosensitive Actuators via Embedded 3D Printing. *Advanced Materials*, 30(15):1706383, 2018.

[144] N. G. Tsagarakis, T. Horne, and D. G. Caldwell. SLIP AESTHEASIS: A Portable 2D Slip/Skin Stretch Display for the Fingertip. In *Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, WHC '05, pages 214–219, Washington, DC, USA, 2005. IEEE Computer Society.

[145] Ultraleap. Leap Motion Controller, 2022.

[146] Ultraleap. Powered by Ultraleap, 2022.

[147] Ryusei Uramune, Hiroki Ishizuka, Takefumi Hiraki, Yoshihiro Kawahara, Sei Ikeda, and Osamu Oshiro. HaPouch: A Miniaturized, Soft, and Wearable Haptic Display Device Using a Liquid-to-Gas Phase Change Actuator. *IEEE Access*, 10:16830–16842, 2022. Conference Name: IEEE Access.

[148] Sidra Waheed, Joan M. Cabot, Niall P. Macdonald, Trevor Lewis, Rosanne M. Guijt, Brett Paull, and Michael C. Breadmore. 3D printed microfluidic devices: enablers and barriers. *Lab on a Chip*, 16(11):1993–2013, 2016.

[149] Stephanie Walker, Uranbileg Daalkhaijav, Dylan Thrush, Callie Branyan, Osman Yirmibesoglu, Gina Olson, and Yigit Menguc. Zero-Support 3D Printing of Thermoset Silicone Via Simultaneous Control of Both Reaction Kinetics and Transient Rheology. *3D Printing and Additive Manufacturing*, 6, April 2019.

[150] Thomas Wallin, James Pikul, Sampada Bodkhe, Bryan Peele, Benjamin Murray, Therriault Daniel, Bryan McEnerney, Robert Dillon, Emmanuel Giannelis, and Robert Shepherd. Click Chemistry Stereolithography for Soft Robots that Self-Heal. *J. Mater. Chem. B*, 5, July 2017.

[151] H. Wang, D. Jones, G. de Boer, J. Kow, L. Beccai, A. Alazmani, and P. Culmer. Design and Characterization of Tri-Axis Soft Inductive Tactile Sensors. *IEEE Sensors Journal*, 18(19):7793–7801, October 2018. Conference Name: IEEE Sensors Journal.

[152] Robert Y. Wang and Jovan Popović. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics*, 28(3):63:1–63:8, July 2009.

[153] Momona Yamagami, Keshia M. Peters, Ivana Milovanovic, Irene Kuang, Zeyu Yang, Nanshu Lu, and Katherine M. Steele. Assessment of Dry Epidermal Electrodes for Long-Term Electromyography Measurements. *Sensors*, 18(4):1269, April 2018. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

[154] Osman Dogan Yirmibesoglu, John Morrow, Steph Walker, Walker Gosrich, Reece Cañizares, Hansung Kim, Uranbileg Daalkhaijav, Chloe Fleming, Callie Branyan, and Yigit Menguc. Direct 3D printing of silicone elastomer soft robots and their performance comparison with molded counterparts. In *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 295–302, April 2018.

[155] Eric M. Young, Amirhossein H. Memar, Priyanshu Agarwal, and Nick Colonnese. Bellowband: A Pneumatic Wristband for Delivering Local Pressure and Vibration. In *2019 IEEE World Haptics Conference (WHC)*, pages 55–60, Tokyo, Japan, July 2019. IEEE.

[156] Yuan-Fang Zhang, Colin Ju-Xiang Ng, Zhe Chen, Wang Zhang, Sahil Panjwani, Kavin Kowsari, Hui Ying Yang, and Qi Ge. Miniature Pneumatic Actuators for Soft Robots by High-Resolution Multimaterial 3D Printing. *Advanced Materials Technologies*, 4(10):1900427, 2019.

[157] Huichan Zhao, Aftab M. Hussain, Mihai Duduta, Daniel M. Vogt, Robert J. Wood, and David R. Clarke. Compact Dielectric Elastomer Linear Actuators. *Advanced Functional Materials*, 28(42):1804328, 2018.

[158] Huichan Zhao, Aftab M. Hussain, Ali Israr, Daniel M. Vogt, Mihai Duduta, David R. Clarke, and Robert J. Wood. A Wearable Soft Haptic Communicator Based on Dielectric Elastomer Actuators. *Soft Robotics*, 7(4):451–461, August 2020.

[159] Huichan Zhao, Kevin O'Brien, Shuo Li, and Robert F. Shepherd. Optoelectronically innervated soft prosthetic hand via stretchable optical waveguides. *Science Robotics*, 1(1), December 2016. Publisher: Science Robotics Section: Research Article.

[160] Gang Zheng, Yuan Zhou, and Mingda Ju. Robust control of a silicone soft robot using neural networks. *ISA Transactions*, 100:38–45, May 2020.

[161] Zheng Wang, Panagiotis Polygerinos, Ali Alazmani, and Alexandre Campo. Fluidic Control Board.

[162] Qian Zhou, Sarah Sykes, Sidney Fels, and Kenrick Kin. Gripmarks: Using Hand Grips to Transform In-Hand Objects into Mixed Reality Input. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–11, New York, NY, USA, April 2020. Association for Computing Machinery.

[163] Mengjia Zhu, Amirhossein H. Memar, Aakar Gupta, Majed Samad, Priyanshu Agarwal, Yon Visell, Sean J. Keller, and Nicholas Colonnese. PneuSleeve: In-fabric Multimodal Actuation and Sensing in a Soft, Compact, and Expressive Haptic Sleeve. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–12, New York, NY, USA, April 2020. Association for Computing Machinery.

[164] Craig B. Zilles and J.K. Salisbury. A Constraint-Based God-Object Method for Haptic Display. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 3, pages 146–151 vol.3, August 1995.