Interactive Perception for Robotic Manipulation of Liquids, Grains, and Doughs

Carolyn Matl

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2021-174
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-174.html

August 9, 2021

Interactive Perception for Robotic Manipulation of Liquids, Grains, and Doughs

by

Carolyn Chen Matl

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering — Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ruzena Bajcsy, Chair
Professor Ronald Fearing
Professor Hannah Stuart

Summer 2021

Interactive Perception for Robotic Manipulation
of Liquids, Grains, and Doughs

Abstract

Interactive Perception for Robotic Manipulation
of Liquids, Grains, and Doughs

by

Carolyn Chen Matl

Doctor of Philosophy in Engineering — Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Ruzena Bajcsy, Chair

With the advent of robotic solutions in both the home and industrial environments comes the need for robots that can dependably perceive and manipulate unstructured or deformable materials like liquids and doughs. Robust handling of these complex materials often depends upon prior knowledge of key material properties such as viscosity or stiffness, which ultimately affect material dynamics. However, these material properties are difficult to observe with traditional sensing mechanisms, e.g. with passive, image-based observations. Furthermore, these materials are generally associated with complex models or high-dimensional representations, which make real-time dynamic predictions intractable.

Interactive perception enables robots to observe and learn from signals that would otherwise not be present. In this work, we develop and leverage non-traditional sensing techniques that are used within an interactive perception robotic framework to estimate physical properties of liquids, grains, and deformable objects. In particular, this research aims to extract key material parameters through direct interaction and observation of induced signals (e.g., sound or force). The estimated parameters are then used to reason about the materials for dynamic robotic manipulation tasks such as precision pouring and dough shaping. We additionally employ simplified representations of these complex materials for use in real-time applications. We hope that this work will highlight the importance of exploring new sensing modalities, which could enable robots to intelligently interact with and manipulate unknown and unstructured objects and materials.

To my parents, Chih-Shan Chen and Tze-Chi Huang, to my kindergarten teacher, Mary Yarusso, and to my love Matthew Matl.

Contents

List of Figures

Acknowledgments

One of the greatest gifts I have ever received from my parents was their unending support in my education. Whenever I felt like I was trudging along on this sinuous, rocky road that is a PhD, my parents were always there to assure me that I was carving this path for myself, to grow more resilient, creative, and independent. They encouraged me to treasure this moment of self-improvement, for this time truly was a treasure that cannot ever be taken away. I am forever grateful for the experiences and opportunities during my PhD that have greatly shaped how I now think, act, and learn. Most of all, I am thankful for the people in my life who have helped me on this road, whether they shined a light for me into the great unknown, held my hand along a daunting path, or were there in the dust with me, paving a way forward.

Of course, I owe much of my PhD education to my advisor Ruzena Bajcsy, who, as my academic mother, not only nurtured my creativity and curiosity, but also fostered a loving and supportive environment that allowed us to flourish in our own unique endeavors. Ruzena and her pioneering work in active perception served as a great source of inspiration, and when I did find myself inspired, she would be ready to support me, no matter how unconventional the idea was. Under her wing, I flew with true academic freedom. I also wish to thank my wonderful thesis committee members Ronald Fearing and Hannah Stuart, who have both been incredibly influential and inspirational with their own research and valuable advice. Not only did Ron teach me how to refine a research paper, but he also taught me how to be a better teacher and mentor. Hannah taught me how to effectively structure my thesis and presentations. Ron and Hannah both cultivated similarly supportive lab environments, and it is no wonder that I was drawn to their research. I additionally would like to thank Dieter Fox and Robert Matthew for taking time out of their busy schedules to serve on my qualifying exam committee. Both during my internship at NVIDIA and throughout my qualifying exam process, Dieter was always ready to ask the important questions. Rob never hesitated to sit down with me to brainstorm an idea or work out a problem. Evidently, I was truly fortunate to have this amazing committee of mentors, who not only have guided me through my PhD, but are also utterly kind people.

Needless to say, I've been blessed throughout my life with countless incredible teachers, each of whom have laid down the stepping stones leading to the path I now trod. In this moment of reflection, my heart especially calls on the loving memory of Mary Yarusso, my Kindergarten teacher, who passed from her long battle with cancer just a few weeks after my thesis presentation. She taught me to be loving, generous, and kind, and she truly was my day-one supporter. She and my first grade teacher Linda Fear have celebrated all of my major life milestones with me, and I will always cherish the fact that I could share the moment of my thesis presentation with both of them. I'd also like to thank my second grade teacher Diane Hackett for enticing me to love math through Starbursts, my art teacher June Adinolfi for teaching me the fundamentals of color and design, and for instilling in me a great amount of creativity and an appreciation for aesthetic figures and slides, and my violin teacher Laura Broverman for showing me that hard work and perseverance were beautiful qualities and, together, could create beautiful things.

I am also indebted to Peter Ramadge, whom, without his early mentorship and encouragement, I would have never found myself in graduate school. As my junior and senior thesis advisor at

Princeton, he taught me how to be meticulous with experiments, how to write a research paper that was engaging yet focused, and, most importantly, how to stay curious and love the unanswered questions. I am also thankful for Tom Funkhouser, my first ever research advisor who was patient enough to debug my code with me, Cameron Chen, my graduate student mentor who showed me that research could be fun and meaningful, Andrew Houck, who convinced me to go to graduate school, Jeffrey Snyder, who inspired my love of engineering new gadgets, and Sharad Malik, who taught me to love teaching and convinced me to apply and come to Berkeley.

Once at UCBerkeley, I was fortunate to have the opportunity to work with many fantastic mentors and collaborators early on, including Ronald Fearing, Roy Fox, Animesh Garg, Ken Goldberg, Aimee Goncalves, Sanjay Krishnan, Michael Laskey, Jeff Mahler, Robert Matthew, Benjamin McInroe, Lauren Miller, Vatsal Patel, Milad Shirani, David Steigmann, and Brijen Thananjeyan. A pivotal moment in my PhD was during my internship at NVIDIA Robotics in Seattle, where by literal luck of the draw, I got an interview opportunity for their research lab. There, I was mentored by several amazing researchers, including Dieter Fox, Renato Gasoto, Miles Macklin, Hammad Mazhar, Yashraj Narang, and Fabio Ramos. I especially would like to thank Dieter and Fabio for taking time on a weekly basis to provide me with insightful feedback, and Yash for being the best mentor I could ever ask for. I only hope that I can follow in their footsteps. As a matter of course, my talented undergraduate mentee Josephine Koe, who was so diligent and resourceful, made my mentorship job quite easy and fun.

Indeed, some of the best mentorship and advice I've been given in graduate school came from none other than the friends I made here. In particular, Andrea Bajcsy, Anusha Nagabandi, and Esther Rolf have been a powerhouse teeming with ideas, suggestions, and advice. Not to mention, they were always there to cheer me up with a silly meme or to indulge in a delicious feast with me. I'm blessed with the best of friends, and have enjoyed sharing meals with new friends I've made along the way in graduate school, including Alon Amid, Elliot Chang, Michael Danielczuk, David Fridovich-Keil, Katie Glockner, Jana Lee, Sammy Sarte, Daniel Seita, Cheebio Walls, and Melvin Walls. Through the close-knit environment the Semiautonomous Seminars fostered, I got to know more amazing friends and mentors from Ron Fearing, Shankar Sastry, Hannah Stuart, and Claire Tomlin's groups, like Dapo Afolabi, Somil Bansal, Jaime Fisac, Sylvia Herbert, Patricia Hidalgo-Gonzalez, Jessica Lee, Ellis Ratner, Dexter Scobee, Laura Treers, and Justin Yim.

I've been blessed by an incredible support network of generous and kind people who were there to help smooth this rocky road. Two wonderful people that were absolutely essential in helping me graduate are Shirley Salanio and Jessica Gamble. In addition to helping me navigate graduate school's countless hoops and hurdles, Shirley was always there to lend an empathetic ear. Up in Cory Hall, we could always depend on Jessica to help schedule a last-minute meeting or to ensure a meeting would run smoothly, and she always made sure we were well fed. Helping me navigate my spiritual hoops and hurdles were Fr. Carl and Fr. David. It was a true blessing to attend RCIA under their direction, and graduate school could have been a lot more stressful if not for the power of prayer. On the topic of road bumps, the preliminary exam was one of the most nerve-wracking periods of my life, and without the support of my prelim friends Andrea Bajcsy, Laura Brink, Sarah Dean, Gautam Gunjala, Keith Moffat, Stan Smith, and Tyler Westenbroek, I'm not sure if I would have had the courage to continue in this program. The PhD can be quite lonely, but I never felt

alone with my WICSE crew, including Coline Devin, Regina Eckert, Kelly Fernandez, Erin Grant, Rachel Lawrence, Caroline Lemieux, Linda Liu, Lydia Liu, Kristina Monakhova, Alyssa Morrow, Mindy Perkins, Rebecca Roelofs, Jaimie Swartz, Elizabeth Yang, and Alice Ye. Friday lunches were always something to look forward to in the company of these amazing ladies. Through my internship at NVIDIA Robotics in Seattle, I had the pleasure of making more lunch buddies. Thanks to Carlos Florensa, Tucker Hermans, Kei Kase, Andrey Kurenkov, Michelle Lee, Tim Lee, Jacky Liang, Ajay Mandlekar, Arsalan Mousavian, Adithya Murali, Yash Narang, Qian Wan, Karl Van Wyk, and many others, I felt right at home in Seattle.

As I am grateful for the new friends who have graced my life, I am also thankful for the many friends that have stayed in touch. To Jayne Bryer, Jojo Bryer, Pam Cai, Tiffany Chang, Danielle Egipto, and Nicole Gagne, thank you for being absolute darlings and bringing sunshine to my life ever since middle and high school. To my college friends, including Paul Von Autenreid, Aliisa Bocarsly, Paul Chang, Lynse Cooper, Elaine Chou, Rishika Dewan, Demi Fang, Lily Feng, Stacey Huang, Ghassen Jerfel, Jean Juang, Max Kaplan, Shreyas Lakhtakia, Xiaohua Liang, Lawrence Liu, Sindiso Nyathi, Brielle Rowe, Matt Silver, Jenny Sun, Siyu Yang, Katherine Ye, Yolanda Yeh, Victor Ying, Sandy Yuan, and Ellen Zhou, thank you for getting me through late night labs in the E-quad, playing and appreciating beautiful music with me, hanging out and playing board games, cooking and savoring delicious food with me, and always reaching out. Let's always stay in touch.

In many ways, I have been blessed by the loving people who support me every day, especially by the loving people I call family. I include in this family my academic family, led by Ruzena and my incredible siblings: Aaron Bestick, Zoe Cohen, Katie Driggs-Campbell, Michael Estrada, Laura Hallock, Isabella Huang, Rob Matthew, Valmik Prabhu, and Sarah Seko. I count myself as incredibly fortunate to have joined the HARTLab with amazing role models and mentors around like Aaron, Katie, Laura, Rob, and Sarah. They all, in their own way, helped nurture a space where I felt respected and supported. And Isabella, although we did not get to travel to Paris as we originally intended, I'm glad I had you as my travel buddy in Macao, land of the egg tarts, crepe cakes, and dim sum. I can't wait for all our future HARTLab reunions.

And finally, to those closest to my heart, thank you. To my grandparents, especially my ah-gongs up in heaven, thank you for all your hard work and for supporting Mom and Dad in chasing the American dream. To my aunties, uncles, and cousins, especially Mei and Nick Avello, Evelyn Chen, Rei and Lily Chiang, Uncle Edwin and Auntie Jane, Uncle Eric and Auntie Grace, Uncle Paul, Auntie Amanda, Uncle Jeff, and Thomas Tang, and Auntie Selina, thank you for checking up on me and taking care of me when I most needed it. To my new family, Lisa, Peter, James, Alexa, Christina, Mary, Anthony, David, and John Matl, Grandma Judy and Grandpa Vance, Papa Chuck and Mama Jane, and Babička and Dědeček, thank you for being my steadfast cheerleaders, for your perpetual love and encouragement, and for Matthew. To Chamois, my fluffy angel, thanks for the laughter and for the indispensable cuddles that got me through the first half of graduate school. To my dearest brother Curtis, thank you for all your phone calls and pep talks to check in. I aspire that one day I may help as many people as you help on a daily. To my parents, thank you for always believing in me, for all the sacrifices you made for my American education, and for your unconditional love. And to Matthew, thank you for everything. You will always be my better half. I thank God for my family and friends, and for all the blessings I could ever wish for.

Chapter 1

Introduction

Aging populations across the globe present an urgency for developing automated solutions for personal care and the home. As a result of this trend, an area of research that has garnered much interest in recent years is in service robots, which are expected to aid humans in tasks such as cleaning and cooking. However, current robots remain challenged in their capability to robustly handle the unstructured and deformable objects and substances that abound in environments like the kitchen. Technical challenges that hinder robotic dexterity of such materials largely lie in the realms of perception, modeling, and control [230, 217].

Interactive perception presents a pivotal step towards unstructured dexterous manipulation; we propose and develop new sensing methods for the handling of these complex materials. Jeannette Bohg, et al., contribute a thorough definition of interactive perception in a review [16]:

> Interactive Perception ... exploits any kind of forceful interaction with the environment to simplify and enhance perception.
>
> — physical interaction creates a novel sensory signal that would otherwise not be present.
> —the prediction and interpretation of this novel signal becomes simpler and more robust.

Interaction *simplifies and enhances perception*. We aim to make use of interaction to *simplify* perception problems and *enrich* what the robot can observe from the world. For humans, perceiving through exploratory interactions is practically second-nature – we squeeze fruits to determine their ripeness, rattle boxes to listen to their contents, and rub cloths to feel their texture. In Jones & Lederman's "Human Hand Function," they note that active haptic sensing is more or less essential for observing identifying properties about an object [80]. These are all examples of how one forcefully interacts with an object or the environment to "induce a novel sensory signal that would not otherwise by present." And the manner in which one interacts with an object (e.g., pressing it), directly connects with the physical property that one seeks to learn about that object (e.g., its compliance) [107]. To emphasize the last section of this excerpt, ultimately, this interaction enables the prediction and interpretation of the signal to be "simpler and more robust."

As Lederman & Klatzky show, humans make these connections almost automatically – exploratory procedures used by human subjects were found to be strongly linked with specific object properties, and these exploratory procedures were determined to be optimal or even necessary for acquiring the desired information [107]. How, then, can robots be equipped with similar perceptual intelligence? In his paper titled "Robotic Perception of Material" [94], Eric Krotkov asks the questions:

> What physical principles relate the acts of manipulation to percepts such as forces, torques, displacements, and vibrations?
> What acts of manipulation best reveal a given material property?

These questions are precisely what have driven the presented research, and they will be revisited throughout this thesis.

To position this thesis in perspective with prior work, we begin by grouping research on robotic interactive perception into roughly three main categories. First, interactive perception can be quite useful for classification purposes. For instance, in [185] and [23], the authors apply different exploratory motions to containers, like shaking or rattling, and use the induced sound signals to classify the container contents. An additional example in [209] demonstrates the use of robotic pushing to categorize objects as either deformable or rigid. These examples illustrate that simple motions, when applied to induce a change of state of the object in question, can generate clear, differentiating signals.

Another focus of interactive perception research is in parameter estimation in order to produce a more accurate model of an object or the environment. One illustrative example is in Eric Krotkov's Whack and Watch experiment [94], where objects were whacked by a pendulum, and their resulting trajectories were observed in order to estimate their mass and coefficient of sliding friction. Other notable work includes [88] and [128], where robotic manipulators interact with articulated objects such as clippers and drawers. Using visual, haptic, and proprioceptive sensing, the authors show how this interaction enables the robot to build a kinematic and dynamics model of the articulated object. Parameters such as the coefficient of sliding friction or the kinematic structure of an object help construct a more accurate representation of the object, which can later aid the robot in planning and control for manipulation.

This leads to the final category of work, in which interactive perception is used to efficiently teach robots how to perform a manipulation task. For instance, in both [111] and [109], the robotic manipulator learns through a short, interactive training phase how to perform contact-rich tasks such as peg-insertion and screwing caps onto bottles. The authors in [108] take a learning-from-demonstration approach for force-based tasks, where human-guided interactions with linear and planar deformable objects enable the robot to learn how to tie a rope or fold a towel. This group of work showcases how interactive perception is inevitable in completing complex manipulation tasks, especially contact-rich tasks. Thus, incorporating interactive perception in the training phase of a manipulation task will improve the robot's effectiveness of performing the task at run-time.

The work in this thesis mostly falls under the category of parameter estimation. However, in contrast with prior literature, we focus on the perception of non-rigid objects and materials that

undergo plastic deformations. Specifically, we study the perception of key material parameters of liquids, grains, and doughs. By extracting these physically relevant parameters, we show how this can be useful to manipulate the complicated material or object using a relatively simple model and control scheme. Thus, we hypothesize:

> Robotic interactive perception of liquids, grains and doughs enables the accurate estimation of identifying properties of the material and subsequently improves the manipulation of that material.

This thesis will address this hypothesis from various perspectives, using different types of perception modalities and representations. The perception of different materials requires unique frameworks and thus the thesis is structured as follows: **Part I** studies liquids, **Part II** studies granular materials and stochastic dynamics, and **Part III** studies plastically deformable solid objects like dough. Given the particular properties we would like to determine about these materials, we ask the questions: what kind of signal can we design to extract this relevant information, and what sort of sensor would be necessary to observe this signal? Therefore, Chapter 2 discusses how haptic perception is used to feel liquids shifting in containers, Chapter 3 illustrates how depth images of static formations can illuminate the dynamics of scattering grains, Chapter 4 demonstrates how audio signals are used to observe the stochastic dynamics of a bouncing ball, and **Part III** develops a series of new variable-stiffness tactile sensors to palpate deformable objects.

This thesis additionally explores interactive perception using three different types of representations. In **Part I**, a derived physics model informs the robot's interactions with the liquid. **Part II** uses simulation to model granular materials and stochastic dynamics in place of an explicit model. Here, interactive perception plays a role in bridging this gap between simulation and reality. Finally, interactive perception is used in **Part III** in a data-driven approach to directly learn a dynamics model of dough and hand interactions, which, due to their complexity, are impractical to model using existing physics simulations that are too slow to adapt for real-time scenarios. The presented robotic interactive frameworks serve to simplify and enhance the perception of liquids, grains and doughs by linking the materials with specific exploratory procedures and sensing mechanisms hypothesized to be highly effective for perceiving the desired properties. Together with the materials' simplified representations, the frameworks demonstrate improved robotic manipulation capabilities of each material via liquid precision pouring, stochastic dynamic predictions, and dough shaping.

Part I

Liquids: Physical Modeling from Haptic Signals

We begin our exploration of interactive perception with a type of material that a robot will inevitably encounter in both the industrial and home settings – liquid. In particular, we are interested in observing liquid inside a container that might not be known or visible, and we aim to estimate the liquid's mass, volume, and viscosity. Revisiting the question of "what kind of interaction and perception would be most useful for observing these material properties," we observe that the liquid's changing center of mass can be directly modeled by deriving a set of explicit equations relating to the state of the container. In the following work, the analytic equations modeling the static and dynamic states of the liquids help unveil clues as to how the robot can effectively perceive the desired properties. The particular revelation made in this work is that the robot can intentionally rotate the container to induce a change in the center of mass. This change in state can subsequently be measured with a force/torque sensor at the wrist of the robotic arm. Thus, the following chapter explores using physical modeling, haptic sensing, and purposeful rotations to liquid inside a container to measure its mass, volume, and viscosity.

Chapter 2

Haptic Perception of Liquids Enclosed in Containers

This chapter is based on the paper "Haptic Perception of Liquids Enclosed in Containers" [132], written in collaboration with Robert Matthew and Ruzena Bajcsy.

Service robots will require several important manipulation skills, including the ability to accurately measure and pour liquids. Prior work on robotic liquid pouring has primarily focused on visual techniques for sensing liquids, but these techniques fall short when liquids are obscured by opaque or closed containers. This chapter proposes a complementary method for liquid perception via haptic sensing. The robot moves a container through a series of tilting motions and observes the wrenches induced at the manipulator's wrist by the liquid's shifting center of mass. That data is then analyzed with a physics-based model to estimate the liquid's mass and volume. In experiments, this method achieves error margins of less than 1g and 2mL for an unknown liquid in a 600mL cylindrical container. The model can also predict the viscosity of fluids, which can be used for classifying water, oil, and honey with an accuracy of 98%. The estimated volume is used to precisely pour 100mL of water with less than 4% average error.

2.1 Introduction

Perception of liquids inside closed containers has been studied for quite some time by the transportation and aerospace industries, where estimating the volume [97] or sloshing [49] of fuel inside a tank can be used to better control trucks or aircraft. With the advent of service robots that can work in the kitchen, this same problem has now resurfaced at a much smaller scale. *How much milk is inside the carton? How thick is the honey inside the jar?* Answering these questions before opening a container can drastically change how a robot might plan to pour a liquid [100, 47, 96, 196].

However, the complexity of fluid mechanics makes it difficult to accurately model and predict the behavior of liquids, and transparent liquids present a challenge for visual sensors. Several

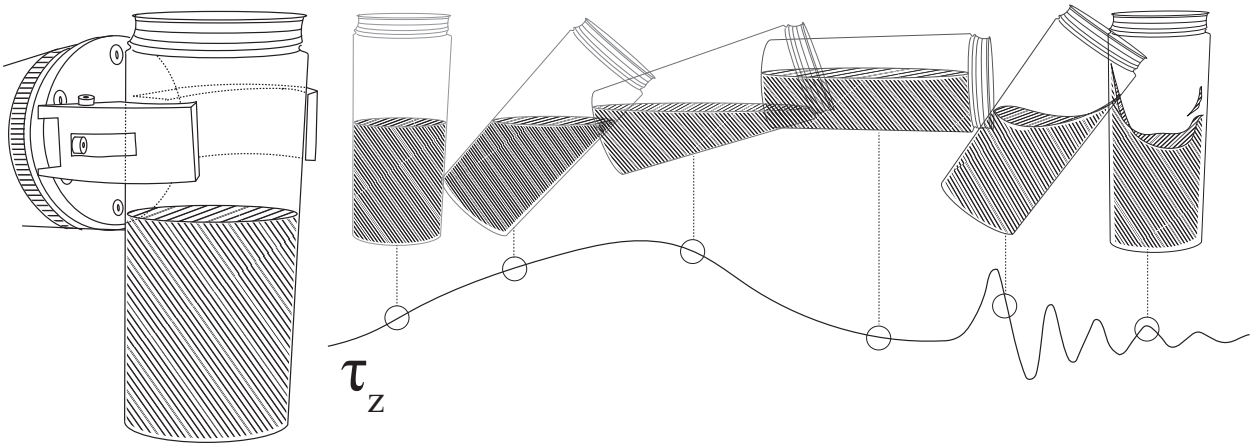


Figure 2.1: Wrench signals at the wrist of a robotic manipulator reflect changes in the center of mass of liquid inside a container. These signals are used to infer key properties of the liquid such as its mass, volume, and viscosity. Rotations are applied to the container in order to observe both static and dynamic behavior of the liquid. © 2019 IEEE

vision-based techniques have addressed this challenge [212, 178, 90], but these methods either only perceive the liquid during the pouring action or do not work when the container is opaque or occluded.

To address these challenges, this chapter proposes a haptic-based method to complement existing visual techniques. Our robot rotates a container filled with an unknown liquid to shift the liquid's center of mass and observes the wrenches induced at the manipulator's wrist during both static and dynamic states of the liquid. The method then analyzes this wrench data with a physics-based model to estimate the liquid's mass, volume, and viscosity. To the best of our knowledge, this is the first work to use robotic haptic sensing and physics-based reasoning to completely determine these three key parameters of liquids inside containers.

The main contributions of this chapter are:

- Time-domain system identification of key parameters of liquids inside containers (mass, volume, and viscosity) using haptic data.
- A physics-based model that utilizes both the static and dynamic states of the liquid for parameter estimation.
- A demonstration of the use of liquid parameter estimation for robotic precision pouring.

2.2 Related Work

Liquid modeling and simulation

The high-dimensional and stochastic nature of fluid flow makes modeling and simulating liquids a challenging and often computationally-demanding task. By narrowing the scope to fluids in a container, it has been shown that basic equivalent mechanical models such as pendulums [43] or multi-mass-spring systems [170] can be sufficiently descriptive for certain objectives. Other work has explored numerical techniques such as the Volume of Fluid method combined with the Navier-Stokes equations to model liquid sloshing [215, 39] or particle-based simulations for tracking liquid flow [144, 212, 179]. A recognized strength of physics-based simulations is their ability to generalize to different liquids, as only few physical parameters such as viscosity need to be changed without the need to relearn a representative model [179]. However, the authors of [179] note that it is still an open challenge to efficiently observe such parameters from real data, which this chapter aims to address.

Active perception and system identification

Inferring model parameters from sensor data is a fundamental focus in the fields of active perception [7] and system identification [5]. The majority of existing literature concerned with estimating liquid properties such as volume or viscosity tackles this challenge by using either special-purpose equipment such as multi-sensor fusion systems built in and around aircraft fuel tanks [97, 64] or camera-based systems that require advanced vision techniques and processing [40, 47]. This chapter proposes a method for finding these parameters by using physics-based models on haptic data.

Interactive haptic perception of objects with internal dynamics

Liquids inside containers are part of the broad category of objects with internal dynamics. It has been shown that for objects in this class, such as articulated objects, [88, 89], direct interactions can allow robots to exploit the object's kinematics and dynamics to improve model information. These interactions can range anywhere from near-static contacts [60] to persistent excitation [23] to derived motions optimized for information gain [176]. In particular, pairing these interactions with haptic sensing enables perception of objects that are unobservable with vision (e.g., liquids in an opaque bottle). In [60, 26], the authors use tactile data during a grasp of a container to distinguish between an empty or full container. A more dynamic motion is applied in [23], in which shaking-induced vibrations are used to classify between different container contents. The angle and frequency of shaking is algorithmically determined to maximize information gain and to infer the viscosity of a liquid inside a container in [176]. While [176] addresses the generation of an action in order to best perceive viscosity, this chapter uses predetermined motions informed by exploratory analysis with a greater focus on parameter estimation.

Manipulation of liquids

Accurate liquid modeling, simulation, and perception is essential for the manipulation of liquids. Slosh-control and pouring are two challenging robotic tasks within the realm of liquid manipulation. Control of tanks or containers while minimizing slosh has applications for ground vehicles, aerospace vehicles, and robotics [49, 170]. Instead of focusing on the control of slosh, the methods in this chapter utilize slosh dynamics to determine the viscosity of a liquid. Numerous papers have also worked on the task of pouring liquids [212, 178, 180, 161, 160, 156, 90], although most rely on vision-based sensing to track liquid flow. Four works of note [174, 180, 90, 115] perform precision pouring, i.e., pouring of specific amounts of fluid. This chapter also attempts this challenging task, but our method differs in that the haptic signals and physics-based reasoning inform the action to take *before* the liquid is poured.

2.3 Problem Statement

Imagine a robotic manipulator that has a stable grasp of an opaque bottle encasing some amount of an unknown liquid. The manipulator is allowed to perturb the bottle in order to haptically perceive the internal dynamics of its contents. Given the wrench forces experienced at the wrist of the manipulator, the goal of this chapter is to determine the mass, volume, and viscosity of the liquid enclosed in the container.

Simplifying Assumptions

It is assumed that the manipulator has already achieved a rigid grasp of the container, maintaining force closure throughout the entire interaction and perception cycle. We limit manipulator motions to rotations of the end-effector wrist joint, and to simplify processing, wrench forces are only measured when the manipulator is not in motion. With these assumptions, wrench forces at the wrist should primarily reflect the shifting center of mass of the liquid. By observing this proxy for aggregate motion of the liquid, the physics models proposed approach the problem macroscopically rather than at the molecular scale. The liquid is assumed to have uniform density and be at room temperature.

Concerning the container, many assumptions were made including that it is rigid and has a known mass and geometry. For much of the analysis in this chapter, we use a cylindrical container and approximate its geometry with a right circular cylinder parameterized by its radius and length. While this geometric assumption enables fast analytic calculations, we can still perform the same analysis on arbitrarily-shaped containers using mesh processing techniques. Knowledge of the grasp point of the container, parameterized by the distance from the bottom of the container to the center of the grasp, is also assumed. Container wall thickness is assumed to be negligible so that the internal volume of the container can be determined from its external geometry, which we acknowledge will induce some small error in our analysis.

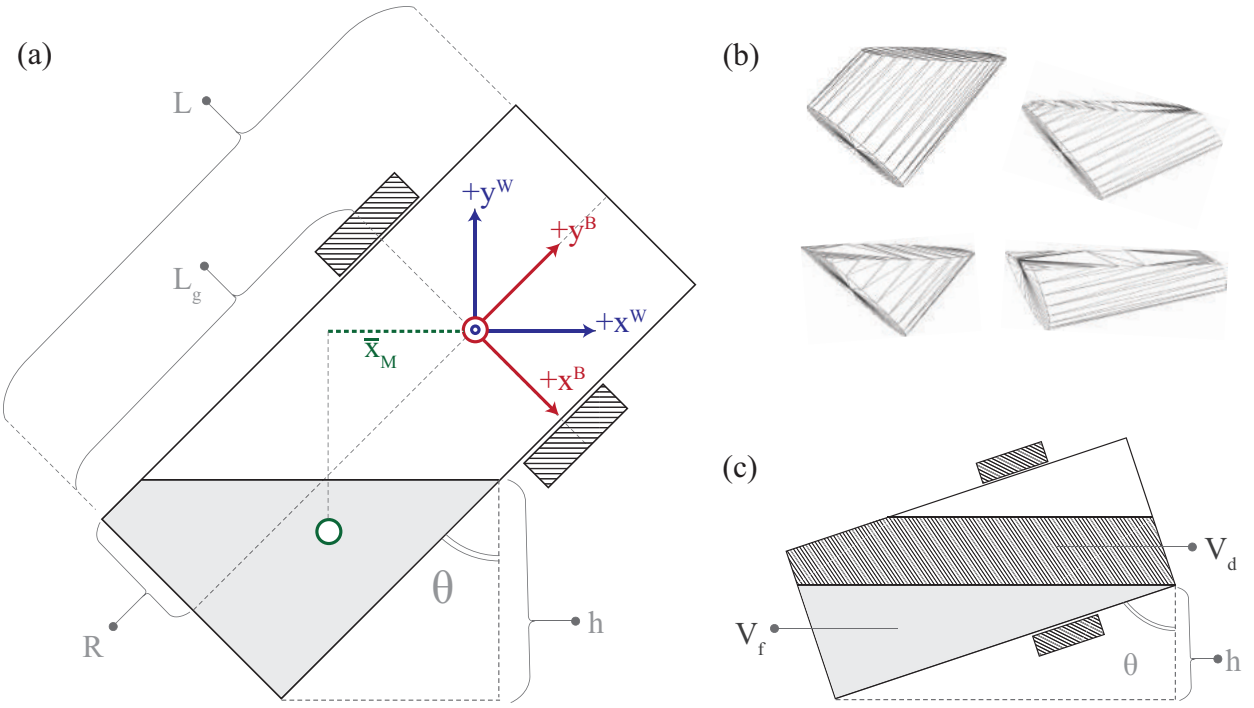


Figure 2.2: (a) A cross-sectional diagram defining variables used to derive parameter estimation formulas. (b) Top to bottom, left to right are the four cases of liquid geometry in a cylindrical container, visualized by the mesh approximation method. (c) A diagram to determine θ for a precise pour. © 2019 IEEE

Definitions

Let the state of the system at time t be defined as $x_t = (f_t, \tau_t, \theta_t)$, where $f \in \mathbb{R}^3$ and $\tau \in \mathbb{R}^3$ are the forces and torques along the x, y, an z axes experienced at the wrist of the manipulator and $\theta \in [0, 2\pi]$ is the clockwise angle of rotation of the manipulator about the z-axis from the vertical. Based on this single-axis rotation constraint, the states of interest for the following experiments are f_x, f_y, and τ_z.

For a cylindrical container, let radius R and length L define its geometry. The robot grasp can be anywhere along the length of the container, and the position of the gripper center relative to the bottom face of the container is defined by the distance L_g. The wrench $F^B = (f^B, \tau^B)$ is originally with respect to the body frame of the system, which we define as rigidly attached to the center of the grasp with positive y pointing towards the top face of the container. The world frame is defined for $+y$ opposite of gravitational force, also with its origin centered between the grasp (see Figure 2.2).

The unknown values we wish to estimate are the mass m_ℓ, volume V, and the dynamic viscosity μ of the liquid. The total mass of the system is $M = m_g + m_c + m_\ell$, where m_g is the known mass of the gripper and m_c is the known mass of the container. Other unknown variables that are useful

for calculations in the following section are the height of the liquid inside the container, h, and the center of mass of the composite system $\bar{x}_M = \bar{x}_g + \bar{x}_c + \bar{x}_\ell$, both of which are defined with respect to the world frame. We use $\bar{x}$ to represent the *horizontal* component of the center of mass. h is defined as the vertical distance from the lowest point of the container to the surface of the liquid. $\bar{x}_g$ is assumed to be 0 since it coincides with the origin of the world frame and $\bar{x}_c$ is assumed to be at the midpoint of the container. $\bar{x}_\ell$ is initially unknown. Figure 2.2 summarizes these definitions in a cross-sectional diagram of the system.

Objective

This chapter aims to solve for m_ℓ, V, and μ given x_t for $t \in [t_0, t_1]$. A predetermined action is taken by the robot, involving a rotation of θ at some velocity $\dot{\theta}$. To simplify analysis, t_0 is defined as the time once this action is completed, and t_1 is a constant time after t_0 (in these experiments, 10 sec). The methods described in the following section are later evaluated by comparing m_ℓ, V, and μ to ground-truth values.

2.4 Methods

Both static and dynamic signals are used to calculate m_ℓ, V, and μ given inputs f_x, f_y, and τ_z. The changing center of mass of the liquid, $\bar{x}_\ell$, is the key to deriving these values. For the following sections, assume the inputs are expressed with respect to the static world frame (see Figure 2.2).

Liquid Mass

Knowing m_c, the mass of the container, finding the mass of the liquid is simple. Given the force in the positive y direction, f_y, of the static world frame, then

$$m_\ell = {}^{f_y}/_{g} - m_g - m_c$$

where g is the gravitational constant $-9.807 {}^{m}/_{s^2}$ and m_g is the measured mass of the gripper in kg.

Liquid Volume

Given the liquid mass from Section 2.4, we can derive an expression for the torque τ_z as a function of the control variable θ and the volume V, an unknown. First, we rewrite τ_z in terms of the centers of mass of the attached system:

$$\tau_z = f_y \bar{x}_M = g(m_g \bar{x}_g + m_c \bar{x}_c + m_\ell \bar{x}_\ell)$$

Because the world frame is anchored at the center of the gripper, let $\bar{x}_g = 0$. We can approximate $\bar{x}_c$ by assuming the container has a shell with a uniform density. Then, relative to the world frame, $\bar{x}_c = ({}^{L}/_{2} - L_g)\sin\theta$. Finally, observe that liquid takes the shape of its container, so $\bar{x}_\ell$ is a function

of rotation angle θ, volume V, and container geometry. Together, we get an expression of τ_z as a function of θ and V:

$$\tau_z(\theta) = g(m_c(L/2 - L_g)\sin\theta + m_\ell \bar{x}_\ell(\theta, V)) \tag{2.1}$$

Let θ represent a vector of rotation angles from a sampled discrete set Θ. This expression allows us to perform nonlinear least squares (NLS) to estimate the parameter V. The objective is to minimize the residuals between our measured values $\tau_z(\theta)$ and our model-based estimated values:

$$V = \arg\min_V \left\| \tau_z(\theta) - g(m_c(L/2 - L_g)\sin\theta + m_\ell \bar{x}_\ell(\theta, V)) \right\|_2^2 \tag{2.2}$$

Below, we describe two different methods that can be used to define the nonlinear function $\bar{x}_\ell(\theta, V)$.

Analytic Model

Many containers can be approximated with simplified geometry (e.g. a rectangular prism or a cylinder). For such containers, we can derive a closed-form analytic expression for $\bar{x}_\ell(\theta, V)$. Here, we define the derived equations for a circular cylinder with radius R and length L and express $\bar{x}_\ell^B$ and $\bar{y}_\ell^B$ with respect to the *body frame*, shifted along the $+y^B$ by L_g. In other words, the origin of the frame of reference lies at the center of the bottom face of the container, with $+x^B$ extending along the radius and $+y^B$ extending along the length of the container.

To make these calculations easier, we first compute the height of the liquid h in the container (see Section 2.3) as a function of V and θ. Because this function is not easily invertible, in practice we use the fact that $h(\theta, V)$ is a monotonically-increasing function in V and find the correct value of h via binary search.

We assume the liquid is at rest, with its surface parallel to the ground plane, and capillary effects are negligible. Depending on h, R, L, and θ, the liquid takes on one of four cylindrical wedge shapes (See Figure 2.2 as illustration). Below, the four cases defining $\bar{x}_\ell(\theta, V)$ collectively form a piecewise continuous function. All formulas are defined for the range $\theta \in [0, \frac{\pi}{2}]$, but through reasoning about symmetry, the full range can be derived.

1. The bottom face is covered, but the top face is dry: This case occurs when $2R\sin\theta < L\cos\theta$ and $h \in [2R\sin\theta, L\cos\theta]$. Then:

$$\begin{aligned}
V &= \pi R^2 \left(\frac{h}{\cos\theta}\right) - 2R\tan\theta) + \pi R^3 \tan\theta \\
\bar{x}_\ell^B &= \frac{\pi R^4}{4V}\tan\theta \\
\bar{y}_\ell^B &= \frac{1}{V}\Big((R\tan\theta \\
&\quad + \frac{h}{\cos\theta} - 2R\tan\theta)^2 \left(\frac{1}{2}\pi R^2\right) + (\tan^2\theta)\left(\frac{1}{8}\pi R^4\right)\Big)
\end{aligned}$$

2. The bottom face is partially covered and the top face is dry: This case occurs when $h \in [0, a]$ for $a = \min[L\cos\theta, 2R\sin\theta]$. We define the following variables to simplify our expression. Let $\alpha = R - \frac{h}{\sin\theta}$ and $\beta = \sqrt{R^2 - \alpha^2}$. Then:

$$
\begin{aligned}
V &= \tan\theta\left(\frac{2\beta^3}{3} - \alpha\left(\frac{\pi R^2}{2} - \alpha\beta - R^2\arctan\frac{\alpha}{\beta}\right)\right) \\
\bar{x}^B_\ell &= \frac{1}{V}\tan\theta\left(\frac{1}{4}\left(\frac{\pi R^4}{2} - \alpha\beta(2\alpha^2 - R^2)\right.\right. \\
&\quad \left.\left. - R^4\arctan\left(\frac{\alpha}{\beta}\right)\right) - \frac{2\alpha\beta^3}{3}\right) \\
\bar{y}^B_\ell &= \frac{\tan^2\theta}{V}\left(\frac{\alpha^2}{2}\left(\frac{\pi R^2}{2} - \alpha\beta - R^2\arctan\frac{\alpha}{\beta}\right)\right. \\
&\quad \left. + \frac{1}{8}\left(\frac{\pi R^4}{2} - \alpha\beta(2\alpha^2 - R^2) - R^4\arctan\left(\frac{\alpha}{\beta}\right)\right) - \frac{2\alpha\beta^3}{3}\right)
\end{aligned}
$$

3. The bottom face is covered and the top face is partially covered: This case occurs when $h \in [a, \cos\theta(L + 2R\tan\theta)]$ for $a = \max[L\cos\theta, 2R\sin\theta]$. The resulting cylindrical wedge is simply a volumetric difference of case (1) and (2). To be precise, let $V_i(\cdot)$, $\bar{x}^B_{\ell_i}(\cdot)$, and $\bar{y}^B_{\ell_i}(\cdot)$ represent the volume and center of mass formulas for case (i) as functions of height $(\cdot)$, with θ implied. Let $h_1 = h$ and $h_2 = h - L\cos\theta$. Then:

$$
\begin{aligned}
V_3 &= V_1(h_1) - V_2(h_2) \\
\bar{x}^B_{\ell_3} &= \frac{\bar{x}^B_{\ell_1}(h_1)V_1(h_1) - \bar{x}^B_{\ell_2}(h_2)V_2(h_2)}{V_1(h_1) - V_2(h_2)} \\
\bar{y}^B_{\ell_3} &= \frac{\bar{y}^B_{\ell_1}(h_1)V_1(h_1) - (\bar{y}^B_{\ell_2}(h_2) + L)V_2(h_2)}{V_1(h_1) - V_2(h_2)}
\end{aligned}
$$

4. The bottom and top faces are both partially covered: This case occurs when $L\cos\theta < 2R\sin\theta$ and $h \in [L\cos\theta, 2R\sin\theta]$. The resulting cylindrical wedge is a volumetric difference of two wedges of case (2):

$$
\begin{aligned}
V_4 &= V_2(h_1) - V_2(h_2) \\
\bar{x}^B_{\ell_4} &= \frac{\bar{x}^B_{\ell_2}(h_1)V_2(h_1) - \bar{x}^B_{\ell_2}(h_2)V_2(h_2)}{V_2(h_1) - V_2(h_2)} \\
\bar{y}^B_{\ell_4} &= \frac{\bar{y}^B_{\ell_2}(h_1)V_2(h_1) - (\bar{y}^B_{\ell_2}(h_2) + L)V_2(h_2)}{V_2(h_1) - V_2(h_2)}
\end{aligned}
$$

As shown, there is not a clean formulation for an expression that maps V to h. Instead, because $V(\cdot)$ is a monotonically increasing function with respect to the variable h, h can be found via binary search for the volume V. After this computation, $\bar{x}^B_\ell$ and $\bar{y}^B_\ell$ are transformed into the world frame.

Mesh Approximation

When a container has more complicated features such as in Figure 2.3, solving for closed-form expressions for volume and center of mass becomes much more challenging. For such containers, we provide an alternative method using polyhedral mass property calculations on mesh approximations [37, 140].

Similar to the method used in [180] and described in [226], we can bisect the container's mesh with a horizontal plane at different heights h and solidify the part of the mesh in the lower halfspace to approximate the shape of the liquid in the container at height h. As in the analytic method, we compute h for a given V by using binary search. The center of mass of the resulting solid is computed using methods described in [140] and implemented in [37]. The analytic and computational mesh methods were verified to produce the same approximate values.

Liquid Viscosity

In contrast with prior sections, viscosity calculations exploit the dynamic modes of the liquid. In particular, we aim to induce either sloshing within the container or laminar flow down its side. For low-viscosity fluids like water, viscosity can be determined by observing the decaying oscillations of the forces caused by the sloshing. For highly-viscous fluids such as honey, the damped response of the forces can be used to approximate viscous fluid flow.

Slosh-induced forces and torques are caused by the shifting center of mass of the liquid inside the container. The slosh mode that dominates the force profile is the fundamental antisymmetric wave, which is characterized by a slosh wave peaking at one end of the container and sinking at the other [43]. In a stationary tank, slosh oscillation decays reflect energy dissipation due to viscous stresses within and at the boundary of the liquid [43]. Several works studying sloshing in a tank have empirically shown that the decay rate of the oscillations is solely a function of tank geometry, fill level, and liquid viscosity. Specifically, [43, 138, 75] define the following. Let $\Delta = \frac{\text{peak amplitude of oscillation}}{\text{peak amplitude of oscillation one cycle later}}$ be the logarithmic decrement of the decaying oscillation. The damping ratio is then defined by:

$$\gamma = \frac{\Delta}{\sqrt{2\pi^2 + \Delta^2}} \tag{2.3}$$

For an upright circular cylindrical tank, the damping ratio of the first symmetric mode is given by:

$$\gamma = 0.79\sqrt{\frac{\nu}{R^{3/2}g^{1/2}}}[1 + \frac{0.318}{\sinh 1.84h/R}(\frac{1-(h/R)}{\cosh 1.84h/R} + 1)] \tag{2.4}$$

where ν is the kinematic viscosity of the liquid, R is the radius of the tank, and h is the fill-level of the tank.

For a container with a different geometry, [75] provides a table of constants C_1 and n_1 corresponding to the characteristic dimension d and the following relation:

$$\gamma = C_1(\frac{\nu}{d^{3/2}\sqrt{g}})^{n_1}$$

This chapter thus aims to induce oscillations for cylindrical containers in the upright position. The input motion we chose is a single impulse rotation from $\theta = \pi/2$ to $\theta = 0$. With γ calculated using Equation 2.3 and h calculated using techniques described in Section 2.4, the viscosity can be calculated by rearranging Equation 2.4:

$$\nu = \gamma^2 \sqrt{R^3 g}[0.79(1 + \frac{0.318}{\sinh 1.84h/R}(\frac{1-(h/R)}{\cosh 1.84h/R} + 1))]^{-2} \tag{2.5}$$

To get the dynamic viscosity, μ, we multiply ν by the inferred density of the liquid ρ.

For highly-viscous fluids like honey, the response is over-damped, meaning no oscillations will occur in the wrench profile. For this case, we estimate the dynamic viscosity μ by approximating the flow once the container is tilted from $\theta = \pi/2$ to $\theta = 0$ as a free surface flow on an inclined plane. [104] defines the mass flux per unit length $\mathcal{Q}$ as

$$\mathcal{Q} = \frac{\rho g \cos\theta h_2^3}{2\mu} \tag{2.6}$$

where $\theta = 0$, ρ is the density of the liquid, and h_2 is the thickness of the liquid. We assume h_2 to equal the height of the liquid at steady state when the container is tilted horizontally ($\theta = \pi/2$). $\mathcal{Q}$ can be approximated by $\frac{\Delta m_\ell}{t_r A}$ where t_r is the 10 to 90% rise time of the damped torque response and $A = \frac{R^2}{2}(\alpha - \sin\alpha), \alpha = 2\arccos(\frac{R-h}{R})$ is the cross-sectional area of the liquid flow sliced parallel to the ground plane. Letting h be the steady state height of the liquid when the container is tilted at $\theta = 0$, we can approximate Δm_ℓ by the mass quantity estimated to move during fluid flow induced by the rotation, or $\Delta m_\ell = m_\ell(\frac{V-Ah}{V})$. Finally, rearranging Equation 2.6, we get:

$$\mu = \frac{\rho g \cos\theta h_2^3}{2\mathcal{Q}}$$

Pouring

Building on the concepts discussed above, we investigate a simple control scheme to pour a specific volume of liquid. Using estimation of the volume V, we can determine a rotation angle θ to pour a specified volume from the container.

Referencing Figure 2.2, let V be the starting volume, V_d be the desired poured volume, and $V_f(\cdot)$ be the final volume of the liquid in the container, corresponding to height $(\cdot)$. We assume that the liquid flow is laminar and that, at the optimal angle θ, V_d is the volume of the liquid above the horizontal plane intersecting the bottom-most point of the container's top face if the container were closed. The objective then is to solve for θ:

$$\theta = \underset{\theta\in[0,\pi/2]}{\arg\min} \left\| V_d - (V - V_f(L\cos\theta)) \right\|_2^2 \tag{2.7}$$

Because V_f monotonically decreases as $\theta \to \pi/2$, a binary search is used to find θ. The pouring experiments only use cylindrical containers, but it is possible to pour from containers of more complicated geometry by using the mesh-approximation method described in section 2.4.

Figure 2.3: From left to right: 48oz Nalgene water bottle used in exploratory analysis, 3 large cylindrical containers used in viscosity estimations, 3 small cylindrical containers used in all experiments, and isopropyl alcohol, coconut milk, canola oil, evaporated milk, mango juice, dish soap, syrup, and honey, arranged in order of approximately increasing viscosity. © 2019 IEEE

2.5 Experimental Setup

Hardware system

All the experiments in this chapter use a Universal Robot UR5 (5 kg payload) robotic arm. While the UR5 has 6 degrees of freedom, for simplicity, the experiments only rotate the 6th joint, or wrist 3, to move the container. An ATI Axia80 EtherNet Force/Torque (F/T) sensor is attached at the end-effector. The six-axis sensor has a 7812 Hz output frequency rate and coincides with the body frame depicted in Figure 2.2. Attached to the F/T sensor is a 3D-printed mechanically-adjusted parallel-jaw gripper. High durability grip tape in the inside of the gripper ensures minimal slippage.

Different types of liquids and cylindrical containers were tested during experimentation, illustrated in Figure 2.3. The different liquids were chosen to represent a range of viscous materials found in a typical kitchen. Different sized and massed cylindrical bottles were used to test their effect on parameter estimation.

Signal processing

The time-series F/T sensor data and UR5 joint data are recorded asynchronously. These are then aligned using system time stamps, down-sampled using a Finite Impulse Response filter, and interpolated so that the F/T and joint position sensors are sampled at the same point in time. F/T sensor offsets measured during a calibration phase are subtracted from the interpolated and down-sampled F/T data. Using the joint position data, a transformation is applied to the six-axis F/T data to convert the wrench from the body frame F^B to the static world frame F^W. Finally, a 1D Gaussian filter with $\sigma = 5$ is applied to all six axes to denoise the data.

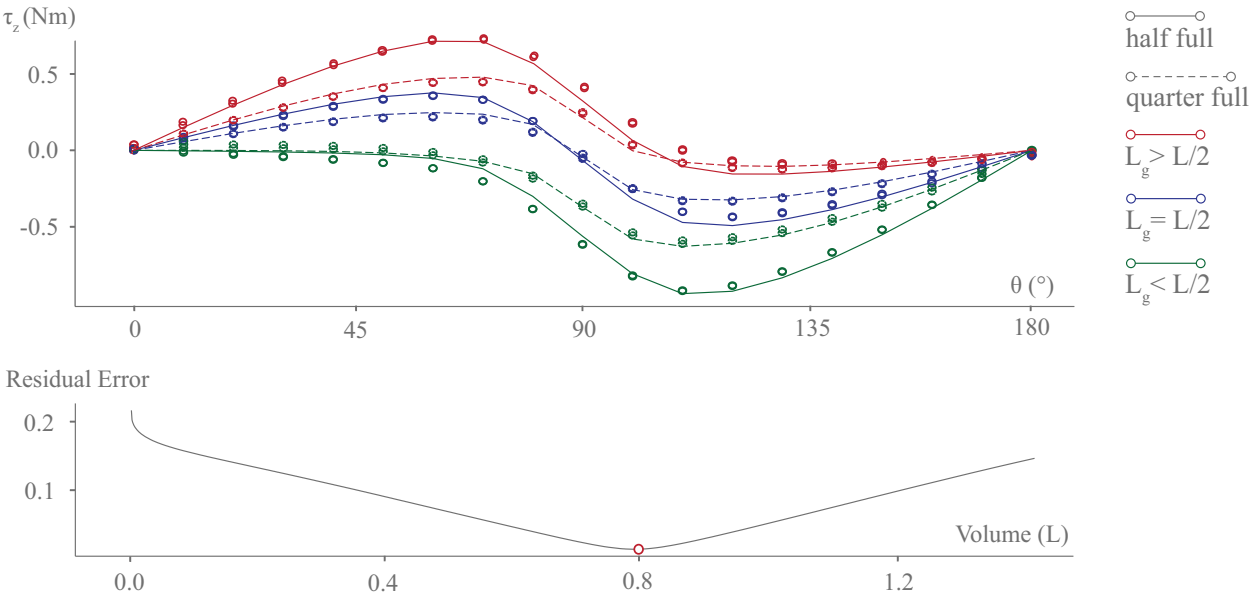


Figure 2.4: (top): Torques measured during a rotation of a half and quarter full Nalgene. Open points represent measured data, and lines represent the predicted torques, demonstrating that our model matches measurements well for different grasp positions and volumes. (bottom): Graph of the residual error between measurements and predictions over the variable V. The convexity of the residual informs us that there should be a global minimum solution. © 2019 IEEE

Exploratory analysis

Prior to experimentation, we performed exploratory analysis on preliminary data. Our first exploration involved filling the $1.5L$ Nalgene with water to $375mL$ and $750mL$ and rotating it at 19 different angles evenly spaced by 10 deg, $\theta \in [0, 180]$ deg. Three different grasp locations were tested – at $L_g = 1/4L$, $1/2L$, and $3/4L$. The resulting data, depicted in Figure 2.4, demonstrates the repeatability of the F/T data and confirmed that τ_z alone was sufficiently informative to infer volume. The rest of the experiments use a grasp at $L_g = 3/4L$ in order to maintain greater stability during pouring. Furthermore, from Figure 2.4, it can be seen that this grasp position has a greater torque response for lower rotation angles, meaning less motion is necessary to extract approximately the same amount of information.

Finally, to ensure that Equation 2.2 could be optimized, we performed a grid search over the variable V. As shown in Figure 2.4, the objective appeared to be convex, so NLS should produce good estimates of the true solutions.

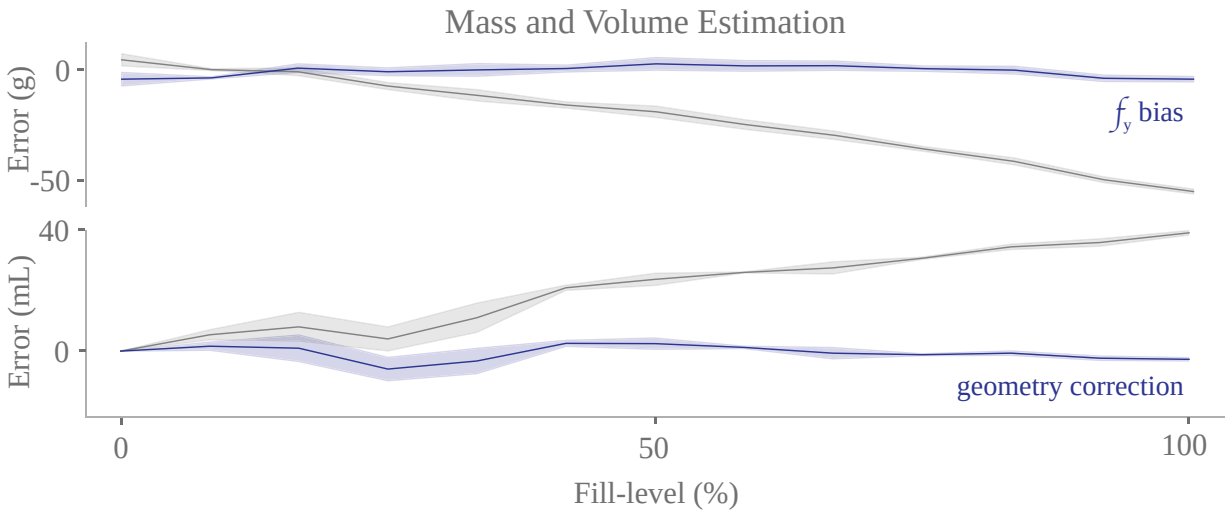


Figure 2.5: Error of mass and volume estimations. Blue curves represent adjustments of f_y measurements and corrections to reflect a better estimate of internal container geometry. © 2019 IEEE

2.6 Results

Predetermined actions are applied to various liquid/container combinations and the accuracy of mass, volume, and viscosity estimations are evaluated below. Note that the mass and volume can be calculated from the same motion. Both calculations assume the liquid is in a static state, so a slow rotation from 0 to 180 deg, stopping at intervals of 10 deg, is applied for these estimations. In contrast, the viscosity estimations rely on observing the dynamic modes of the liquids, so a fast rotation from 90 to 0 deg is applied. Finally, for precision pouring, the applied rotation is computed using the estimated volume. Please see the accompanying video for visualizations of the following experiments: . For visualizations of the following experiments, please see: `https://youtu.be/Qr0B09iQEcM`

Liquid Mass Estimation

We begin with liquid mass estimation by conducting a sensitivity experiment with varying fill-levels of a container. The small cylindrical container depicted in Figure 2.3 is used, which has an external radius and length of 35 and 172 mm, respectively, with a total measured capacity of 662mL. We sampled 13 fill-levels of water, evenly spaced from 0 to 100 percent full, and measured the ground-truth mass. For each fill-level, we performed five iterations of rotating the container from $\theta = 0$ to 180 deg, stopping every 10 deg to let the liquid settle. f_y is calculated for each instance by averaging over the transformed f_y^Ws throughout the entire motion. As shown in grey in the top-left plot of Figure 2.5, there is a gradual increase in the error of our mass estimate as the fill-level increases. This error could be attributed to a miscalibrated conversion from strain-gauge

measurements to forces. Because the error appears to grow linearly with an increased mass, we fit a linear regressor to "re-calibrate" measured f_y forces. In the same figure in blue, we see that this re-calibration helps center the error of the mass estimation at an average error of -0.23g with a 95% confidence interval of ± 0.73g.

Liquid Volume Estimation

The same motion is used for liquid volume estimation. With the data from Section 2.6, we can observe the empirical error on our estimation of volume for different fill-levels of the small cylindrical container. The sampled measurements of τ_Z at different values of θ along with our estimate of m_l allow us to search for a volume V that minimizes the error between the measurements and our calculated τ_Z curve using Equation 2.1. Our first attempt resulted in a large average error, aggregated across all fill-levels, of 87 ± 53mL. However, a closer look revealed that this large error was due to the edge case of $m_l = 0$, for which the objective was not convex. Because we use the shifting center of mass to determine V, there is no way of discerning an empty container from a full container. Thus, our optimization consistently estimated $V = L\pi R^2$ when $\theta = 0$. However, knowledge of the container mass disambiguates the value of m_l, so we can bias the estimator to assume $V = 0$ if $m_l = 0$.

From Figure 2.5, we see that, with this bias, volume estimation performs with an error of 21.46 ± 3.28mL. Notice that our calculations tend to overestimate the actual liquid volume, as expected since we assume internal container geometry is equivalent to external geometry. If we reason about the geometry of the container's cavity, e.g., walls could be $1mm$ thick, then we see that this lowers the 95% confidence interval of the error to -0.65 ± 0.83mL. All 65 estimates of volume were within 13mL of error. Lower volume estimates tended to have greater variance, perhaps due to higher sensitivity to sensor noise. Note that while we tested mass and volume estimation on water, the physics-based calculations will work for any liquid as long as measurements are recorded when the liquid is in steady state.

Using NLS optimization allows for an efficient search of the volume estimate, but to verify that it indeed is finding the global minimum of the objective function, we evaluated the objective with a fine grid search over all possible volumes. Taking the global minimum solution found via grid search, the error in the volume estimate was essentially equivalent to that of using the more efficient NLS method, at a 95% confidence of -0.62 ± 0.82mL. The error-per-fill-level graph is visually identical to the one produced via NLS so it is omitted from Figure 2.5.

Finally, we tested liquid volume estimation using the mesh approximation method. A half-gallon milk carton (total capacity equivalent to about 1.9L) was measured and modeled as a 3D mesh. The milk carton was filled to 250mL, 500mL, and 750mL with water, and five iterations of tilts and measurements were performed per each fill level. Using the 3D mesh of the milk carton, the 95% confidence error margin of the volume estimation optimization was 1.69 ± 24.69mL.

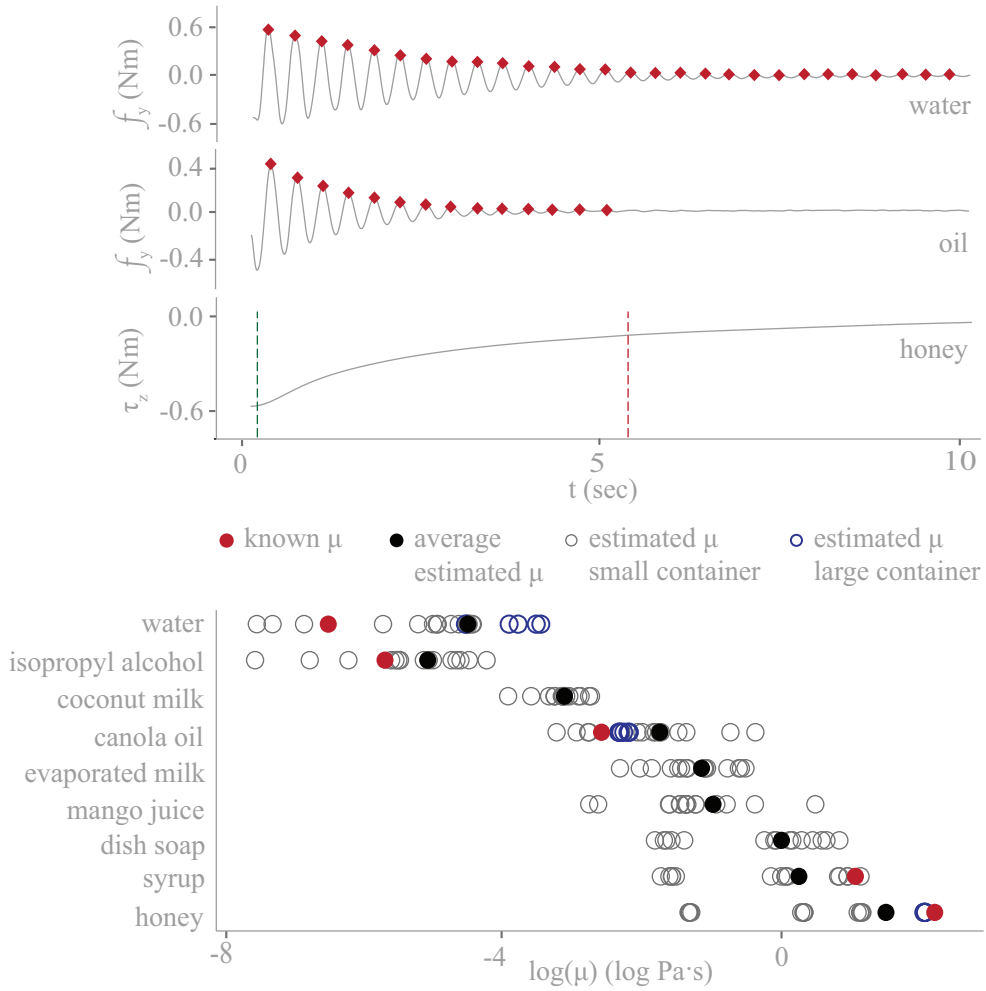


Figure 2.6: (top) Real f_y and τ_z data (filtered and smoothed) after a $90 \rightarrow 0$ deg rotation of water, oil, and honey in the large containers. Detected peaks and rise times used in viscosity calculations are shown. (bottom) Viscosity predictions for the 9 different liquids, arranged from top to bottom with increasing true viscosities. True values of five liquids are denoted by red. Black circles mark the averages of the estimated viscosities per liquid. Note, for illustrative purposes, viscosity estimates are shown on a log scale.

Liquid Viscosity Estimation

We apply a different motion to the container here in order to observe the liquid in its dynamic mode. Instead of a slow rotation, we use a fast rotation from $\theta = 90$ to 0 deg to act as an impulse to the system. The dynamics can be observed via both f_x^W and τ_z^W. If the resulting signals contain oscillations, the algorithm calculates μ via the slosh-dynamics analysis described in Section 2.4. Otherwise, the overdamped response is used to calculate μ via the fluid-flow equation (2.6). We

tested this method of viscosity estimation on nine different liquids – water, oil, honey, isopropryl alcohol, coconut milk, evaporated milk, mango juice, dish soap, and syrup. Each liquid was poured into the small cylindrical container at 137, 292, and 442 mL and a series of five impulse rotations were applied in each scenario. To test the method's sensitivity to varying geometries, we also perform five impulse rotations to 636.11 mL of water, oil, and honey, each in a larger cylindrical container depicted in Figure 2.3.

As illustrated in Figure 2.6, while the variance of the viscosity estimates is high (note, however, on a log scale), the method manages to capture relative viscosity differences between the liquids. In fact, using the predicted viscosities of data from volumes of 292, 442, and 636 mL, a linear SVM classifier with 5-fold cross-validation is able to distinguish between the four lowest viscosity liquids at an accuracy of $78 \pm 11\%$ and between water, oil, and honey with an accuracy of $98 \pm 10\%$. These predictions improve if we narrow the training set to larger quantities of volume. This suggests that Equations 2.4 and 2.6 serve as reasonable parameterizations of the haptic signal τ_z. Furthermore, the method generalizes well to data captured using the larger container. For instance, estimates of the 636 mL of honey had an error of 0.197 ± 0.096 Pa·s when honey's true viscosity is approximately 10Pa·s.

The predominant limitations of the reported method that likely contributed to the wide variance in viscosity estimations are low signal-to-noise ratios (SNR)s of the haptic data as well as imperfect models. Many of the signals collected using the small container had low SNRs, often detecting one or two peaks before the signal was obscured by sensor noise. In contrast, with the larger container and quantity of liquid, the changes in the liquid center of mass were much clearer, as evidenced in signals shown in Figure 2.6. Subsequently, viscosity estimates made from larger containers tended to have a lower variance than those made based on observations with the smaller container. To combat this issue, a more dynamic-sensitive tactile or haptic sensor could be used, paired with an intelligent selection of shaking motions to improve SNR, similar to the active learning framework demonstrated in [176]. As expected, the models used to estimate viscosity were imperfect. For example, the clustering of the estimates of honey are illustrative of the sensitivity of the proposed model to changes in geometry and volume. A better model is needed to overcome these sensitivities.

An Application Demonstration: Pouring

To demonstrate the applicability of the methods described in this chapter, we attempt to pour a specific amount of liquid given estimations of liquid volume. In contrast with other precision-pouring papers, this work pre-computes an open-loop control strategy (specifically, the rotation angle θ) based on the estimated volume. While the proposed control scheme is not very sophisticated, it achieves reasonable effectiveness in this challenging task using only the haptically-estimated volume. Furthermore, this pre-computed rotation angle can act as a guide when paired with closed-loop control schemes, which can serve to dynamically fine-tune or adjust the angle.

We begin by characterizing the robot's pouring precision. Using Equation 2.7 and perfect knowledge of the liquid volume, we attempt to precisely pour a quarter, half, and three-quarters of 200, 400, and 500 mL of water from the small cylindrical container, with 5 trials for each case.

At first, all attempted pours were approximately 50mL below the desired poured volume. After observing a few pours, it was evident that there were two factors contributing to this experimental error. The first is that there was a lip to the cylindrical container, where the radius of the open top was actually significantly smaller than the external radius. To fix this error, we assume that computer vision techniques can be used to estimate the internal radius, wall thickness, and lip dimensions of the container. This is a reasonable assumption as the lid would be off, exposing the true internal geometry of the container. The second factor contributing to error was the surface tension of the water. It was so strong that water would bunch up at almost 5mm above the bottom-most point of the lip, contributing to a significant difference between our model prediction and experimental results. To combat this, we added a term to represent the added height required due to water tension. With these changes, our robot became very precise and repeatable. Under perfect knowledge, the robot operates with an error of at most ± 5 mL poured, with an average error of 1.55 ± 1.60 mL.

We then attempted to precisely pour 100 mL from 200, 300, 400, and 500 mL of water based on the estimated volume of the water, with 5 trials for each case. Here, we make no assumption of the density of the liquid and only base our calculations off of estimates of V. As before, we found that the precision of pouring was incredibly sensitive to the a priori knowledge determining the container geometry. With knowledge of internal container geometry, the original volume estimated is more accurate (5.09 ± 2.59mL) which subsequently allows for a more precise pour. For a 100 mL pour over 20 trials, the robot achieved an average error of 1.80 ± 2.04mL, where the largest error poured was $+9$mL. These results are comparable to related work that uses visual reasoning to precisely pour liquids. Finally, we attempted to pour oil and honey, but these both caused a giant mess and so we decided to postpone those experiments for the time being. However, a reasonable next step would be to incorporate estimated viscosity data for different liquids to guide the *speed* at which the pour is performed for minimal sloshing and maximal efficiency.

2.7 Conclusions

In this chapter, we show that physics-based analysis of haptic signals can achieve high precision estimation of the mass and volume of liquids in a cylindrical container and provide a framework for estimating fluid viscosity. To demonstrate the efficacy of these methods, we performed precise pouring of water where the estimated volume determined the rotation angle θ. In future work, we imagine that incorporating real-time haptic and visual control will improve precision during pouring. Furthermore, we believe the precision of our viscosity estimates can be improved by combining better physics-based parameterization with active perception to optimize for motions to induce higher SNRs of the haptic data. More accurate viscosity measurements could then be used to inform a control strategy to quickly pour viscous fluids with minimal sloshing [196]. Finally, we hope this work can be extended to haptic perception and control of granular materials and more non-newtonian fluids in containers. In the following chapter, we begin our investigation into this realm of problems through the perception and physical parameter estimation of granular materials.

Part II

Grains and Stochastic Dynamics: Simulation Calibration with Depth Images and Sound

In **Part I**, explicit models of a material like liquid revealed how the robot might interact with the material to extract key features like volume or viscosity. For some materials, however, there may not yet exist sufficiently expressive models. Furthermore, the dynamic behavior of the material may require stochastic modeling. Take as an example a container of grains. Due to the material properties of the individual granular particles, we can no longer assume, unlike with liquids, what the granular material may look like at steady state after a rotation is applied to the container. Thus, in **Part II** of this thesis, we dive into the perception of stochastic dynamic systems, specifically, of granular materials in Chapter 3 and a single, bouncing ball in Chapter 4.

First observe that material properties have a direct effect on the collision dynamics of a material. The goal then is to design an interactive perception framework that induces these collision events in an effort to estimate material properties from the resulting signals. However, explicitly modeling the dynamics of granular materials and other stochastic dynamic systems can be rather difficult and complex. In place of analytic equations to represent the material, **Part II** turns to using numerical simulation. State-of-the-art work on both the physics of granular materials and bouncing balls also use numerical simulation as a tool for studying their behavior. However, the relevant material properties are often hand-tuned to match real-life observations. In contrast, the work presented in **Part II** uses real sensor observations to directly tune the simulator to match real-life stochastic dynamics.

The underlying method that acts as the bridge between the simulation representation and reality is a likelihood-free Bayesian inference framework called BayeSim [169]. This framework approximates the posterior of simulator parameters given real observations. This turns out to be quite valuable for this section's problem formulation, where some level of uncertainty over the material parameters of interest is desirable in order to account for the stochasticity observed in real life. Chapter 4 builds on this concept further by directly approximating a posterior over a stochastic parameter.

Chapter 3

Inferring the Material Properties of Granular Media

This chapter is based on the paper "Inferring the Material Properties of Granular Media for Robotic Tasks" [133], written in collaboration with Yashraj Narang, Ruzena Bajcsy, Fabio Ramos, and Dieter Fox.

Granular media (e.g., cereal grains, plastic resin pellets, and pills) are ubiquitous in robotics-integrated industries, such as agriculture, manufacturing, and pharmaceutical development. This prevalence mandates the accurate and efficient simulation of these materials. This work presents a software and hardware framework that automatically calibrates a fast physics simulator to accurately simulate granular materials by inferring material properties from real-world depth images of granular formations (i.e., piles and rings). Specifically, coefficients of sliding friction, rolling friction, and restitution of grains are estimated from summary statistics of grain formations using likelihood-free Bayesian inference. The calibrated simulator accurately predicts unseen granular formations in both simulation and experiment; furthermore, simulator predictions are shown to generalize to more complex tasks, including using a robot to pour grains into a bowl, as well as to create a desired pattern of piles and rings. Visualizations of the framework and experiments can be viewed at `https://youtu.be/OBvV5h2NMKA`.

3.1 Introduction

A granular material is a collection of discrete macroscopic particles that primarily experience inelastic collisions and are usually unaffected by temperature [77]. Naturally, this definition spans a wide variety of materials, from sand and stones to powders and pills to seeds and cereals, making granular material one of the most manipulated substances in the construction, pharmaceutical, agriculture, and food industries [171]. As robots continue to permeate these industries, they will inevitably interact with granular materials in a variety of settings, such as pouring flour in the kitchen, scattering seeds on a farm, or traversing loose terrain [112].

Despite the ubiquity of granular materials, much of the physics behind their complex behaviors remains a mystery. Granular materials are often considered their own category of matter (separate from solids, liquids, and gases), and consequently, new constitutive laws must be defined. While recent work has delivered promising advances in the formulation of continuum models for granular materials [81, 3, 45, 155, 85, 86], researchers frequently rely on numerical techniques and simulations to corroborate and extend their predictions.

The state-of-the-art numerical technique for simulating granular materials is the discrete element method (DEM), where each particle is represented independently. Although DEM has primarily been used in physics, it has also been leveraged by the robotics community for applications such as locomotion on sand or gravel [127, 92, 154] and motion planning for pouring [38]. However, calibrating the material parameters of the simulated grains is still an open problem. Most DEM calibration methods involve a tedious cycle of performing small-scale materials tests, manually tuning parameters, and iterating until the desired experimental behavior is captured. To reduce this burden, the physics community has posed the following challenge: how can particle-scale parameters (e.g., frictional properties) be determined from observations of macroscale behavior (e.g., granular pile shape) [58]? Answering this could lead to more efficient methods for refining simulations to match real macroscale behavior.

This chapter addresses this question by integrating a framework for likelihood-free Bayesian inference (BayesSim [169]) with a fast and efficient physics simulation platform capable of DEM simulations (NVIDIA's Isaac Simulator [157]), as well as experimental observations using off-the-shelf depth cameras. Using this software and hardware framework, depicted in Figure 3.1, particle-scale material properties (i.e., coefficients of friction and restitution) of granular materials are inferred from simple, inexpensive observations of macroscale behavior in the real world. The calibrated simulator can then be leveraged to predict unseen granular formations in both simulation and experiment. This chapter concludes by demonstrating how this research may be applied to robotics tasks by using the developed framework on an industrial robot to manipulate agricultural grains (i.e., couscous and barley).

The main highlights of this chapter are

- A software and hardware framework that integrates likelihood-free Bayesian inference, an efficient physics simulator, and depth images of granular formations to estimate particle-scale material properties
- A demonstration of the use of this framework to calibrate material properties of agricultural grains (couscous and barley) for a simulator, which is then evaluated against new scenarios in simulation and experiment
- A robotic demonstration that uses the proposed framework to reason about agricultural grains in order to complete a simple manipulation task

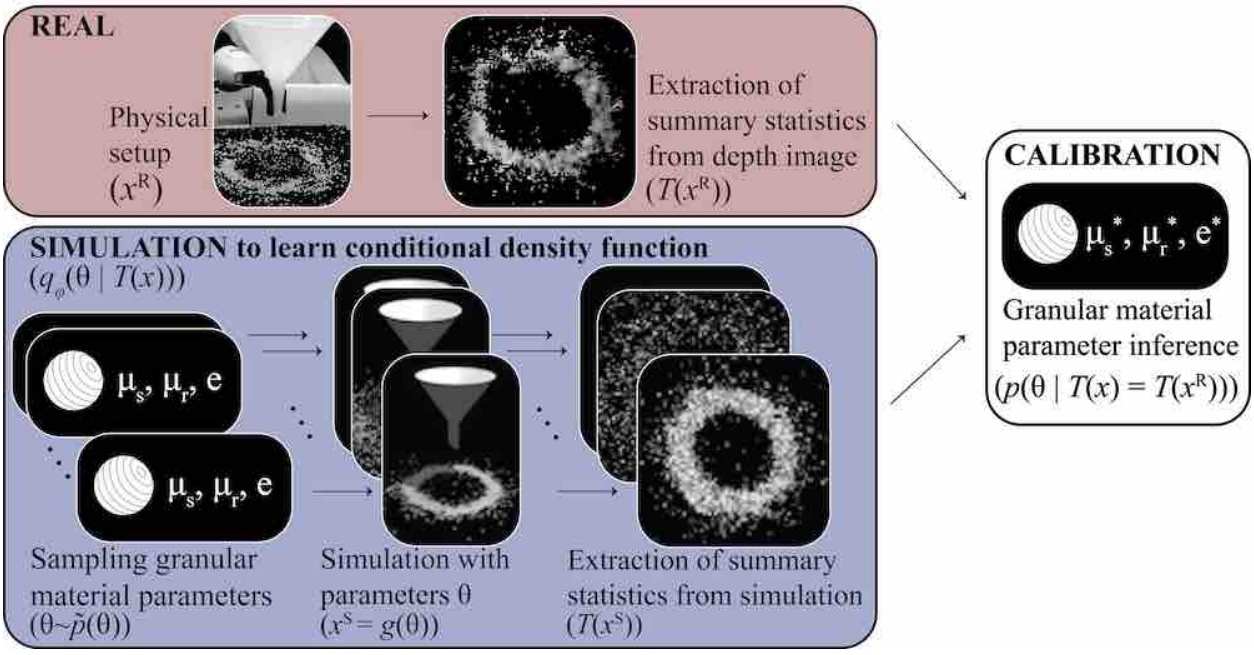


Figure 3.1: Granular parameter inference framework. (Bottom branch): Multiple granular parameter sets are randomly sampled from a prior distribution $\theta \sim \tilde{p}(\theta)$, and for each, summary statistics $T(x^S)$ are extracted from a simulation $g(\theta)$. The set of $T(x^S)$ are used to learn a conditional density function for the parameters using BayesSim. (Top branch): The summary statistics of a depth image of a real granular formation are extracted. With the conditional density function, a posterior is estimated to infer the likeliest simulation granular parameters to match the macroscopic granular behavior.

3.2 Related Work

Granular Materials in Physics

The mechanics of granular materials is an active area of research within the physics community. Recent advancements present new constitutive relations of granular media that relate flow behavior to particle-scale properties like grain size [86], geometry [135], and surface friction [87]. Throughout this body of work, DEM simulators are consistently used to evaluate and extend theoretical explanations [36, 227, 87, 81, 135]. This emphasis on simulation encourages the development of frameworks that can infer difficult-to-measure material parameters (e.g., frictional coefficients of a grain) from macroscale behavior of collections of grains (e.g., pile shape). This work proposes one such framework, designed primarily for robotic applications due to its speed, efficiency, and minimal hardware requirements.

Granular Materials in Robotics

Within robotics, the mechanics of granular materials has been investigated to design jamming grippers and manipulators [19, 200, 118, 29], as well as mobile robots that can traverse granular terrain [232, 112, 127, 92, 228, 154, 202]. Recent work has explored the perception of granular materials [31, 23, 185] and their manipulation in scooping [31] and pouring [181] tasks. Such work has largely focused on the classification of granular materials rather than estimation of material parameters; furthermore, it has often been highly task-specific. However, granular parameter identification is important to enable physical reasoning of how different grains may behave in unseen scenarios while dynamically interacting with robots. Due to the complexity of granular physics, researchers in robotics have leveraged simulation to model robot-granular interactions [232, 228, 127, 154, 202].

Simulation of Granular Materials

The state-of-the-art numerical technique to simulate granular materials is the Discrete Element Method (DEM), where each grain is generally represented as an independent particle that interacts with others only through direct contact. Other granular simulation techniques include the Material Point Method (MPM) [70, 45], which constructs a background mesh to compute forces, and Position Based Dynamics (PBD) [145, 124, 126], which directly manipulates particle positions to enforce attachments and resolve collisions, typically trading off accuracy for speed. Although accurate, DEM often uses explicit time-integration, which can require impractically small time steps for stiff systems. In this work we model contact as a nonlinear complementarity problem (NCP) which is solved implicitly using a GPU-optimized non-smooth Newton solver in the NVIDIA Isaac Simulator platform [157, 125]. This enables both high accuracy and rapid simulation of $\sim$2000 spherical grains. Furthermore, the platform allows direct input of particle-scale parameters such as geometry and material properties, as well as a programmatic interface in which new testing scenarios can be rapidly created. The platform has recently been applied to reinforcement learning applications for non-DEM simulations [22, 116].

Automatic Calibration of Granular Material Simulation

Most techniques used to calibrate the material parameters for DEM simulation require tedious experiments and manual tuning until the desired experimental behavior is captured [32]. Thus, recent work aims to automate this process. For instance, in [41], a genetic algorithm and direct optimization are used to determine the frictional coefficients of sand grains by measuring discharge time from an hourglass. In [13], an artificial neural network is used to infer multiple material parameters of iron ore sinter; the network greatly increases the speed of inference at the expense of training time. Recently, likelihood-free inference techniques, such as Approximate Bayesian Computation (ABC) methods, have been explored to increase the interpretability of results, as well as to enhance robustness to non-unique mappings between material properties and macroscale behavior.

Specifically, in [24], an iterative Bayesian filtering framework was used to infer the coefficients of friction of glass beads based on 100 measurements of porosity and stress per iteration.

In the present work, we leverage a likelihood-free Bayesian inference framework called BayesSim [169]. The framework allows the complexities of the simulator to be fully abstracted from the inference process and requires only a single training phase followed by a prediction step. This is in contrast to multiple prediction steps necessary in typical Bayesian filtering. Because BayesSim is an ε-free method [162] (i.e., it does not rely on sampling with acceptance region defined by ε), it is generally more sample-efficient and produces more accurate parametric approximations of the posteriors than ABC methods [162]. Furthermore, the preceding studies measured macroscopic behavior using highly-specific experimental setups and advanced laboratory equipment, which may be prohibitively expensive for many robotics applications. In contrast, the present work measures macroscopic behavior (e.g., granular formations from pouring grains through a funnel) using single images taken with inexpensive, off-the-shelf depth cameras.

3.3 Problem Statement

Simplifying Assumptions

This section defines the key simplifications made in the simulator. First, we consider grains with a length scale of at least 1 mm, thus excluding powders. Given that the average size and mass of a grain can be measured, they are assumed as known inputs to the simulator. While natural grains will represent a distribution of sizes and masses, we make the simplification that all grains are of the same size and mass.

We further assume that the grains are perfectly rigid, cohesionless, and experience negligible drag, so that the only sources of energy dissipation are friction and collisions with each other and the ground. The forces experienced during these interactions, depicted in Figure 3.2, are functions of the coefficients of restitution e, sliding friction μ_s, and rolling friction μ_r for the grain-to-grain or grain-to-ground interactions. To simplify parameter search, we assume that the grain-to-grain and grain-to-ground coefficients are equivalent. Together, these three parameters enable the simulator to generate a diversity of realistic granular behavior. Using a subset of $\{e, \mu_s, \mu_r\}$ restricts the simulator's expressiveness and may likely cause the real and simulated distributions of granular formations to be completely distinct.

Finally, we represent all grains as spheres. While it has been shown that aggregate behavior is dependent on grain geometry and size, this coarse approximation makes collisions easier and faster to compute, and thus practical for robotic applications. Further, although the parameters are bounded by real-world values, it is not the goal of this chapter to estimate physically-exact friction and restitution coefficients. Rather, the calibration framework aims to accurately simulate macroscopic behavior of the granular material. For non-spherical grains, larger or smaller valued material parameters may sufficiently compensate for a spherical approximation.

Definitions

Let $\theta = [\mu_s, \mu_r, e]^T$ represent a vector of the grain simulation parameters, composed of the coefficients of sliding friction μ_s, rolling friction μ_r, and restitution e. Let X represent the birds-eye-view depth image of the grains once they have reached steady-state. Variables X^r and X^s are used to distinguish real from simulated depth images, respectively, and $X^s = g(\theta)$, where $g(\theta)$ is the simulation. Because of the high-dimensional space of depth images X^r and X^s, we define the random variable $T(X)$ as a vector composed of statistics (such as average distance of grains from the centroid) to summarize the observed state X. Let the likelihood function be defined as $p(T(X)|\theta)$ and the posterior be defined as $p(\theta|T(X) = T(X^r))$. The likelihood function, defined implicitly by $g(\theta)$, is often intractable to evaluate or unavailable with such complex simulations, thus motivating the use of a likelihood-free inference approach.

Objective

Given samples of $T(X^s)$ and the corresponding parameters used to generate the simulations, we would like to approximate the posterior function $p(\theta|T(X) = T(X^r))$ such that, from a real observation X^r, we can infer granular simulation parameters θ from the macroscopic behavior of the material, summarized by features $T(X)$. Because DEM simulations leverage various numerical solvers and are associated with complex systems of differential equations, we choose a simulator-agnostic inference framework that approximates the intractable posterior by sampling from the simulator. The likelihood-free Bayesian inference framework BayesSim [169], discussed in the following section, is used to approximate this posterior.

3.4 Methods

The goal of this chapter's proposed framework is to estimate granular parameters $\theta = [\mu_s, \mu_r, e]^T$ from summary statistics $T(X^r)$ using BayesSim, which approximates the posterior $p(\theta|T(X) = T(X^r))$ by sampling from a simulator $g(\theta)$. In this section, we discuss the BayesSim likelihood-free framework, simulator $g(\theta) = X^s$, physical experimental setup to observe X^r, and extraction of summary statistics $T(X)$.

Granular Parameter Inference Using BayesSim

BayesSim directly learns a conditional density function, $q_\phi(\theta|T(X))$ (where ϕ is the set of learnable parameters of q), by mapping summary statistics to a mixture of K Gaussian components via random Fourier features [168]. The Mixture Density Random Fourier Feature (MDRFF) network learns from N pairs of parameters and summary statistics, $\{\theta_i, T(X_i^s)\}_{i=1}^N$, generated by independently sampling from a predefined prior distribution, $p(\theta)$, and forward-simulating each sample, θ_i. For twice differentiability of q, the Matérn kernel is used with $\nu = 2.5$, where ν increases with smoothness. The posterior is then parametrically approximated using the learned conditional density $q_\phi(\theta|T(X))$ and a real observation $T(X^r)$. The main advantages of using BayesSim over

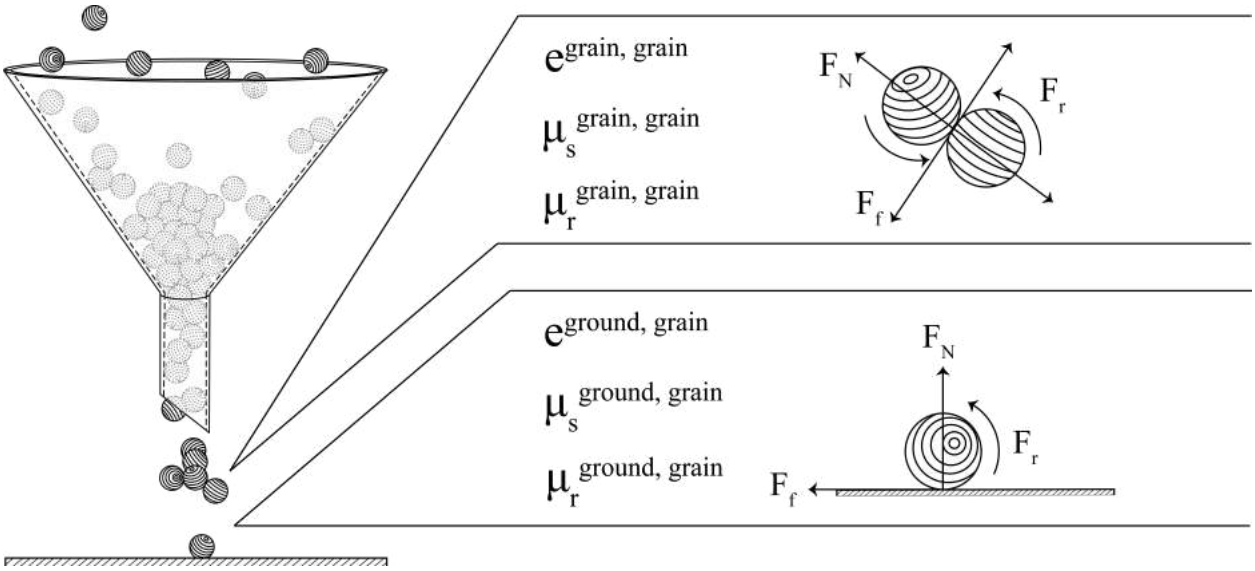


Figure 3.2: The simulations capture sliding friction forces (F_f), rolling friction forces (F_r), and energy loss due to inelastic collisions for grain-grain and grain-ground interactions. These forces are parameterized by the coefficients of sliding friction (μ_s), rolling friction (μ_r), and restitution (e), respectively. © 2020 IEEE

classical optimization methods are that the complexities and inaccuracies of the DEM simulator and PDE solver are incorporated in the estimated posterior. The multimodality and uncertainty of the approximated posterior can be used to analyze the accuracy with respect to the parameters θ and robustness with respect to measurement noise. In addition, because BayesSim is an ε-free method, it improves on ABC methods by being more sample efficient and producing more accurate posterior approximations [162] with a limited number of samples. After the initial cost of running N simulations and training the model, BayesSim can quickly produce posterior approximations, making it a compelling method for granular material inference problems in robotic applications.

Simulation

Critical dimensions are made equivalent in the simulated and physical scenes. The simulated funnel, which was modeled in 3D based on the physical funnel, is positioned at an equivalent distance above the ground surface. The spherical grains are scaled to match the diameter of a single grain of Israeli couscous along its longest axis (4 mm). For simplicity, at the start of the simulation, the 2000 grains are each spaced at 8 mm intervals in a $10 \times 10 \times 20$ grid, where the bottom grains are just below the top of the funnel.

The coefficient of restitution e is the ratio of a grain's speed before and after a collision; this coefficient ranges from 0 to 1, corresponding to perfectly inelastic and elastic collisions, respectively. The ranges of the coefficients of sliding friction μ_s and rolling friction μ_r that create a noticeable

diversity in granular formations are 0 to 1 and 1×10^{-7} to 1×10^{-1}, respectively. For this reason, rolling friction is sampled from a logarithmic distribution. A simulated depth camera is positioned at a birds-eye viewpoint, 29 cm from the surface ground, giving it a 58 cm $\times$ 58 cm field of view.

The solver chosen is NVIDIA Isaac's Preconditioned Conjugate Residual (PCR) solver, and 20 inner iterations with 4 outer iterations are used in experiments [125]. The simulator runs with time step $1/60$ sec, 10 substeps, and a relaxation factor of 0.75 for stability and convergence.

Physical Setup

For the physical experiments, the funnel is positioned with its bottom tip 12 cm above the ground surface. The surface is covered in thin, moderate-friction black velvet to limit the spread of the grains upon impact, as well as to simplify segmentation. The primary grain used in the experiments is Israeli couscous, which has the approximate geometry of a rounded cylinder; barley, which has a similar density but is slightly larger and more ellipsoidal, is also used in the latter half of the experiments. An Intel RealSense D435 depth camera is positioned at 40 cm above the ground surface. This position was chosen to balance the trade-off between field-of-view and clarity of the depth image. Although the raw depth images are still noisy at 40 cm, eliminating the background using color segmentation results in well-defined point clouds of the granular formations. An ABB YuMi dual-arm robot is used for the robot experiments.

Summary Statistic Extraction

A majority of DEM calibration work relies on extracting specific features (e.g., angle of repose) of a conical pile in order to infer a single parameter (e.g., μ_s). However, initial experiments showed that pouring grains can result in piles or rings, which may be more informative of other material parameters. Moreover, for shallow formations, it may be difficult to measure angle of repose from a depth image. We take a more generalizable approach and propose using a vector of 16 summary statistics to serve as $T(X)$.

To extract the summary statistics features from a granular formation, the depth image X is first downsampled by factor n. For X^s, $n = 20$ and for X^r, $n = 10$ to match resolutions. Because the X^r are fairly noisy, a color threshold mask is used to segment pixels corresponding to the granular formations. These pixels are then reprojected into 3D using the camera intrinsics matrix. For the real depth image, a plane is fit to reprojected pixels corresponding to the ground surface. The orientation of the fitted ground plane informs the centering and re-orientation of the granular point cloud to account for camera tilt. Let r represent the vector of radial distances of grain points in the horizontal plane from the centroid of the granular formation and z be the vector of grain point heights.

The summary statistics of a single depth image $T(X)$ are chosen as

$$[max(z), \mu(z), \sigma(z), max(r), \mu(r), \sigma(x), \sigma(y), \sigma(r),$$
$$IQR(x), IQR(y), IQR(r), KURT(r), dCor(r,z), df, b, A]^T$$

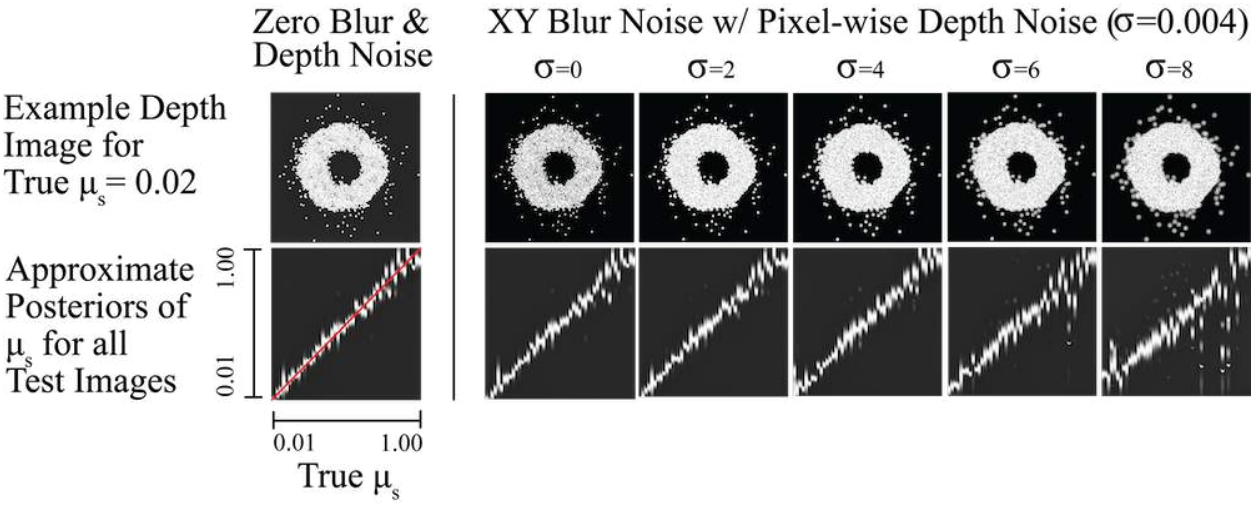


Figure 3.3: Sensitivity to observation error. An example X^s is shown perturbed by different levels of noise. 50 different test images X^s were generated by evenly sampling $\mu_s \in [0.01, 1.0]$. The resulting approximated posteriors are plotted per true value of μ_s, where the highest peaks correspond to the brightest points. A perfect inference would result in a white line overlapping the red. While the model is fairly robust to the level of per-pixel Gaussian depth noise applied, accuracy is more affected by Gaussian blur. Posterior peaks become wider, and distributions can become multimodal with $\sigma > 4$. © 2020 IEEE

where $\mu(\cdot)$, $\sigma(\cdot)$, $IQR(\cdot)$, $KURT(\cdot)$, and $dCor(\cdot,\cdot)$ are the mean, standard deviation, interquartile range, kurtosis, and distance correlation. A chi-distribution, chosen for its likeness in shape to a cross-section of a granular formation, is fit to r. Statistics df, b, and A denote degrees of freedom, shift, and scale, respectively, from a standardized chi probability density. These statistics were chosen to summarize the dispersion and statistical dependence of the grain locations. A preliminary ablation study suggested that subsets of $T(X)$ were sufficiently descriptive in capturing the granular distributions, unless the subsets contained less than three features or the features exclusively pertained to either r or z.

3.5 Sensitivity Tests for System Design

Preliminary Tests for System Noise

When running the simulator to train BayesSim, there are two main sources of uncertainty. The first is purely algorithmic, due to the non-determinism of the simulator and numerical error. To measure algorithmic uncertainty, the simulator was run 1000 times with identical conditions except for sliding friction, where $\mu_s \in \{0.01, 1\}$, evenly split. The measured summary statistics for each μ_s had variances of at least two orders of magnitude smaller than their mean values.

Second, an insufficient number of samples for training and interpolation uncertainty may contribute to poor performance. To test these, a training set was created by sampling μ_s uniformly between 0.01 and 1. A 50-simulation testing set was generated by sampling between 0.01 and 1, independently of the training set. Supporting intuition, the accuracy of the friction coefficients inferred by BayesSim improved with more samples. For a training set of 500 samples, the error in the friction coefficient estimates is 0.0191 ± 0.0799, while for 1000 samples, the error is reduced to 0.0062 ± 0.0018.

Sensitivity to Observation Error

A key assumption of our system is that the summary statistics of the real and simulated depth images will lie within the same distribution. However, the two might not align or overlap given drastically different observations (e.g., due to a low-quality experimental depth image). To test the effect of observation noise on inference, we simulate two types of camera noise on a 50-sample testing set, where μ_s is evenly sampled from 0.01 to 1 with all other parameters equivalent. An XY Gaussian blur with $\sigma = \{0, 2, 4, 6, 8\}$ is convolved with the depth image, and pixel-wise Gaussian noise with $\sigma = \{0, 0.001, 0.002, 0.003, 0.004\}$ is added to each depth value, with ranges chosen to mimic realistic noise magnitudes. These two perturbations are applied independently, and the resulting posteriors per true friction value are illustrated in Figure 3.3. As shown, inference of μ_s is fairly robust to pixel-wise noise, but is sensitive to high values of Gaussian blur.

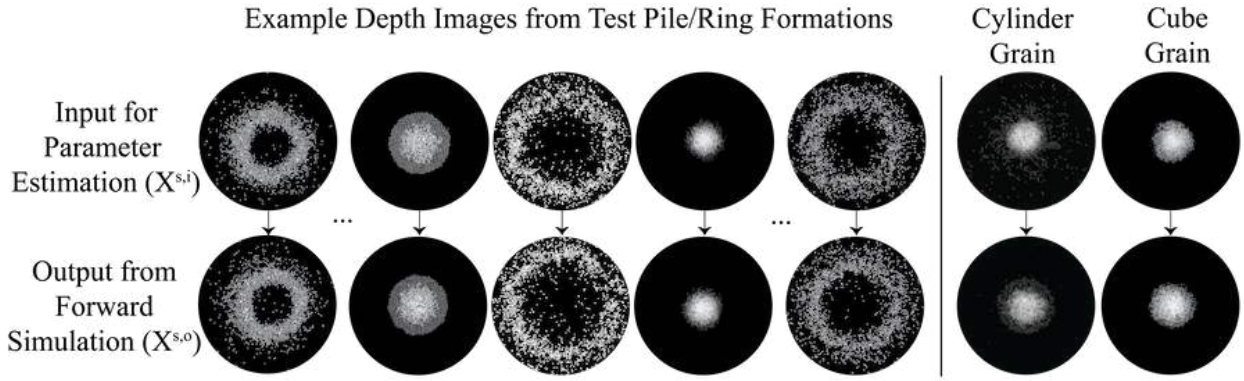
Propagation of Error

Given an observation X, the approximated posterior encodes the uncertainty of the inferred parameters θ^*. Sensitivity tests were performed to investigate how this uncertainty propagates to errors in the summary statistics $T(g(\theta^*))$. Fixing all other parameters, μ_s, $\ln(\mu_r)$, and e required errors of ± 0.05, 3, 0.01, respectively, to perturb the height and average radial distance to more than the width of a single grain. For real-world context, manufacturing applications may require that the diameter of a granular pile be predicted to within 5% of the true value. (For example, for a pile with base diameter equal to 50 times a grain diameter of 4 mm, an error of 1 cm may be allowable). With the proposed system, this specification is fulfilled by the optimal values, with allowable variation of ± 0.1, 4, 0.05 that is within the performance achieved in the following experiments.

3.6 Results

Sim-to-sim Parameter Estimation

The first objective was to test the accuracy of the trained BayesSim model on simulated depth images. Given a simulated input image $X^{s,i} = g(\theta)$, we aimed to quantify how close the inferred parameters $\theta^* = \mu_s^*, \mu_r^*, e^*$ were to θ. Furthermore, to evaluate the propagation of error in θ^*, a forward simulation was run to obtain a new pile $X^{s,o} = g(\theta^*)$. The difference between the piles was



Error in μ_s	-0.0025	0.0395	0.0046	0.1331	0.1209	Not applicable	
Error in μ_r	-4.37E-7	-1.67E-7	1.81E-9	-5.17E-6	1.38E-8	Not applicable	
Error in e	-0.0029	0.1143	-0.0114	0.0211	0.0226	Not applicable	
$\|T(X^{s,i})-T(X^{s,o})\|_2$	0.3334	3.1038	1.3444	0.8788	0.7655	3.2383	1.5713

Figure 3.4: Examples comparing simulated depth images of testing set and simulated depth images from running a forward simulation with inferred parameters. Table shows corresponding errors in inferred parameters, as well as L2 errors between summary statistics. © 2020 IEEE

evaluated via the L2-norm error between $T(X^{s,i})$ and $T(X^{s,o})$, both standardized over the summary statistics of the training dataset. Note that, for some applications, evaluating particular subsets of the summary statistics might be more useful (e.g., height of the granular pile or the angle of repose). To enable generality, all 16 summary statistics defined in Section 3.4 are used to calculate the L2-norm error. We avoid evaluating accuracy via direct comparisons of the downsampled depth images, as the computed error would be highly sensitive to misalignments of the scattered grains rather than reflect meaningful differences in spatial distributions.

The 50 test grain formations were simulated by randomly sampling μ_s and e uniformly and μ_r logarithmically from their given ranges. The average error of the inferred e was -0.0008 ± 0.0148, while that of μ_s and $ln(\mu_r)$ were 0.0380 ± 0.0646 and -1.1695 ± 0.7032, respectively. As in Section 3.5, these error bounds correspond to negligible changes in the resulting statistics. As shown in Figure 3.4, forward simulations of the inferred parameters for the 50 test formations generate visually equivalent shapes. To consider as a baseline for the following experiments, the average L2-norm error between the 50 pairs of $\{T(X^{s,i}), T(X^{s,o})\}$ was 1.582 ± 0.3356.

Sim-to-sim comparisons also enable a principled approach to testing sensitivity to model error. For instance, the grains are spherical, when in reality most grains are not. We tested sensitivity to mismatched geometry by creating a testing set of cylindrical and cube-shaped grains, where the diameter and length of each grain were 4 mm, the diameter of the training set spheres. Each testing set consisted of 50 different formations which varied in μ_s, while e and μ_r were fixed at 0.5

and $1E{-}6$, respectively. Parameters μ_s, μ_r, and e were inferred from the test formations and forward simulated with spherical grains. Comparing $T(X^{s,i})$ (with cylindrical and cubic grains) with $T(X^{s,o})$ (using spherical grains) the L2-norm errors were 4.0139 ± 1.0574 and 1.8301 ± 1.9275, respectively. Spherical grains can approximate formations made with cubes better than cylinders, possibly due to the asymmetry of cylinder geometry. Depending on the application, a spherical representation may be sufficient.

Real-to-sim Parameter Estimation

The goal of this experiment was to test whether material parameters of real-world granular formations could be inferred to produce accurate simulations of macroscopic behavior. The initial test set was composed of real depth images of ∼2000 grains of Israeli couscous, poured through a funnel fixed at 12 cm above the ground surface. Ten different pours were executed, and for each, ten consecutive depth images were collected and averaged to mitigate temporal noise from the camera. For each of the averaged images $\{X^{r,i}\}_{i=1}^{10}$, summary statistics $\{T(X^{r,i})\}_{i=1}^{10}$ were computed, and μ_s, μ_r, and e were inferred. Forward simulation was executed for each parameter set, and the L2-error was computed between the summary statistics of the real and forward-simulated granular formations. From the ten parameter sets inferred, the five corresponding to the lowest errors were averaged. The inferred parameters for couscous were $\mu_s = 0.6687$, $\mu_r = 8.1506 \times 10^{-7}$, and $e = 0.7689$, and the parameters for barley were $\mu_s = 0.3807$, $\mu_r = 1.0613 \times 10^{-6}$, and $e = 0.4792$.

Forward simulation with the estimated parameters captures similar macroscopic behavior as the five real-world pours at 12 cm, with an average L2-norm error of 2.2263 ± 0.1377 and 2.0311 ± 0.3807 for the couscous and barley formations, respectively. However, it is critical to test how well these parameters generalize to pours at different heights. Five more pours of each grain were performed, with each pour at a different height: $\{2, 4, 6, 8, 10\}$ cm off the ground. For each of these pours, a forward simulation was run with the same grain parameters while shifting the funnel height. Figure 3.5 illustrates that the inferred parameters generalize reasonably well, with relatively low L2 errors. This suggests that it may be possible under this framework to reason about pour heights to create desired pile and ring shapes given a specific type of grain. We test this hypothesis in the following experiment.

Robotic Demonstrations

The first demonstration evaluated whether the presented framework could enable an industrial robot to pour granular material into a desired shape, as may be necessary in a kitchen or factory. A new BayesSim model was trained on 1000 simulated piles. The simulator parameters μ_s, μ_r, and e were set to their inferred values for couscous (0.6687, 8.1506×10^{-7}, and 0.7689, respectively), and the funnel height was uniformly sampled between 1 and 13 cm. Two summary statistics were computed from the resulting granular formation (i.e., the 5th and 50th percentile of granular radial distance, which approximated the inner and outer radii of a ring). BayesSim was then evaluated on a testing set to infer the funnel height from these two summary statistics; the height was estimated within 0.0530 ± 0.2221 cm. For the demonstration, summary statistics were chosen to correspond

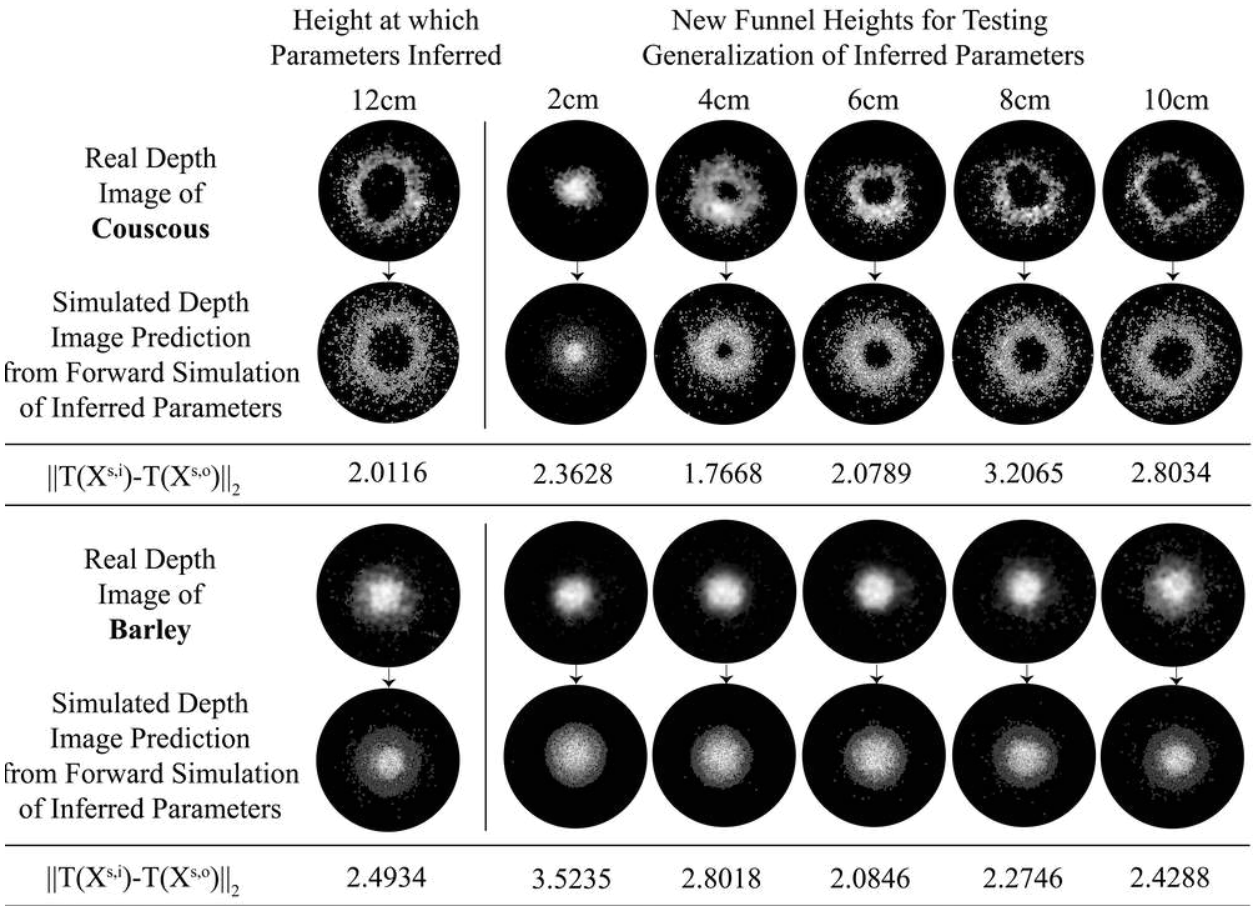
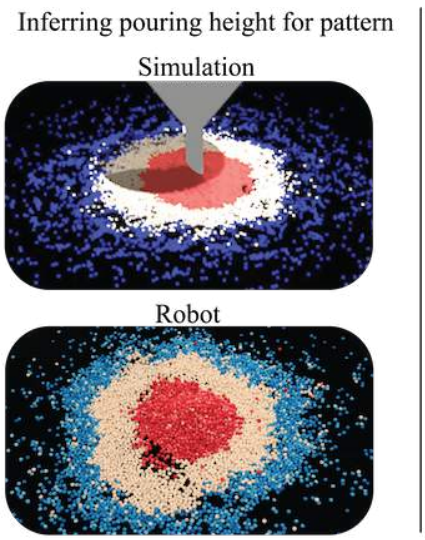
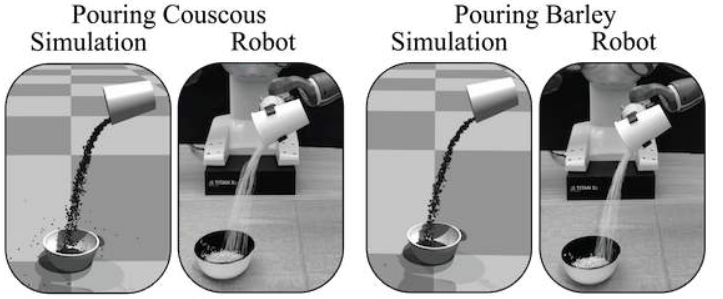


Figure 3.5: Comparisons of real depth images of piles poured at different heights and simulated depth images generated from running a forward simulation at the corresponding heights, using parameters inferred from a height of 12 cm. L2 errors are listed below the depth images. © 2020 IEEE

to a desired pattern of two concentric rings with an inner pile. BayesSim inferred corresponding funnel heights of $\{27.1, 10.1, 1.5\}$ cm. The robot was commanded to pour couscous from these heights, resulting in the desired pattern (Figure 3.6).

The second demonstration evaluated how closely the presented framework could predict undesirable spilling during a granular pouring task. The calibrated simulator was used to simulate the pouring of couscous and barley into a cereal bowl, with the simulation relaxation constant increased from 0.75 to 0.9 to ensure accurate simulation of high-speed collisions. The number of grains that spilled out of the bowl was counted. The industrial robot was then used to precisely repeat the experiment in the real-world, pouring grains into a velvet-lined cereal bowl to match simulator conditions. As tabulated in Figure 3.6, in most cases, the calibrated simulator predicted the number of spilled grains across different pouring heights with surprising accuracy. (Please see





Number of grains that bounce out at different heights (3 pours each)

15cm	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
30cm	23, 20, 18	12, 15, 8	1, 0, 0	3, 2, 3
45cm	52, 50, 53	46, 36, 44	4, 12, 10	22, 11, 24
60cm	133, 90, 116	115, 95, 127	35, 48, 49	33, 40, 42

Figure 3.6: Two granular manipulation tasks were tested. (Left): The height of the funnel is inferred to create a desired ring shape. Performance is demonstrated by creating a pattern of three concentric rings. (Right): The calibrated simulator is used to recreate a real-life scene of pouring grains into a bowl. At various heights, both the simulations of couscous and barley reasonably estimate the number of grains that leave the cereal bowl. © 2020 IEEE

the supplementary video at `https://youtu.be/OBvV5h2NMKA`.)

3.7 Conclusions

In this chapter, the material parameters of granular materials were inferred using a new framework combining likelihood-free Bayesian inference, efficient simulation, and simple experiments. The use of GPU-based simulation and off-the-shelf depth cameras may be particularly appealing for robotics applications. Simulation-to-simulation inference was highly accurate, and simulation-to-experiment inference trailed closely in performance. Robotics demos showed that the inferred parameters generalized well to different pouring heights, and furthermore, that a robot can effectively reason about granular material to pour desired granular formations and predict its behavior in dynamic scenarios.

There are numerous exciting opportunities for future work in perceiving and reasoning about granular materials. To improve inference, observations can be amended to include temporal sequences of depth images, with additional summary statistics to capture transient dynamics. To enhance simulation, NVIDIA's Isaac Simulator can be applied to model cohesion, allowing liquid-solid interactions (e.g., capillary bridging); in addition, simulated particle geometries could be represented as superquadrics [165], enabling more accurate approximation of complex real-world grain geometries. Furthermore, more elaborate robotic manipulation tasks can be explored, such

as moving the end effector along a trajectory to create an asymmetric grain trail, or scooping and pouring granular material into and out of assorted containers. These robotic tasks involve substantial interactive forces with the granular media, which can be modeled using promising techniques such as Granular Resistive Force Theory [202]. Ultimately, we believe similar parameters of granular materials can be extracted through the rich audio signals resulting from collisions between the grains and their environment. We begin tackling this problem in the following chapter, where we explore extracting physical parameters from audio signals of a single bouncing ball.

Chapter 4

Sim-to-Real from Sound for Stochastic Dynamics

This chapter is based on the paper "STReSSD: Sim-to-Real from Sound for Stochastic Dynamics" [134], written in collaboration with Yashraj Narang, Dieter Fox, Ruzena Bajcsy, and Fabio Ramos.

Sound is an information-rich medium that captures dynamic physical events. This chapter presents STReSSD, a framework that uses sound to bridge the simulation-to-reality gap for stochastic dynamics, demonstrated for the canonical case of a bouncing ball. A physically-motivated noise model is presented to capture stochastic behavior of the balls upon collision with the environment. A likelihood-free Bayesian inference framework is used to infer the parameters of the noise model, as well as a material property called the coefficient of restitution, from audio observations. The same inference framework and the calibrated stochastic simulator are then used to learn a probabilistic model of ball dynamics. The predictive capabilities of the dynamics model are tested in two robotic experiments. First, open-loop predictions anticipate probabilistic success of bouncing a ball into a cup. The second experiment integrates audio perception with a robotic arm to track and deflect a bouncing ball in real-time. We envision that this work is a step towards integrating audio-based inference for dynamic robotic tasks. Experimental results can be viewed at `https://youtu.be/b7pOrgZrArk`.

4.1 Introduction

The sensory modality of sound can impart practical physical intuition of the surrounding world. In particular, sounds arising from the dynamic interaction of objects or substances through direct contact are highly informative. These sounding interactions enable humans to accomplish various system identification tasks, such as discerning the size [59], length [21], and material [91] of a struck object. The abundance of information encoded in event-driven sound alone calls for the development of robotic techniques to extract physical understanding from this evidently descriptive modality.

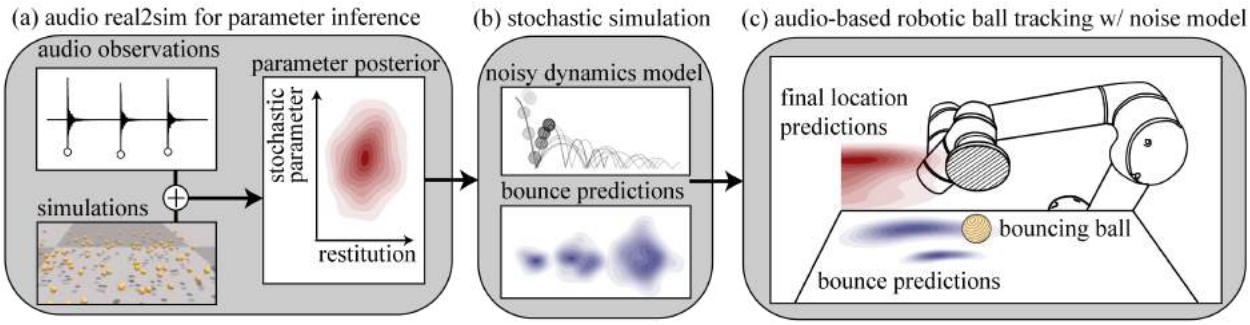


Figure 4.1: STReSSD is a framework that uses sound to bridge the sim-to-real gap for a stochastic dynamic process (e.g., a bouncing ball). (a) Simulation is used to learn a conditional density function relating a material and noise parameter to collision times and positions, and posteriors over these parameters are generated from audio observations of bouncing balls. (b) A probabilistic dynamics model is learned via the calibrated stochastic simulator. (c) This model is used with real-world auditory feedback for a reactive robotic ball-tracking task.

Thus, auditory perception presents a natural sensory choice to infer properties of and track an object that is dynamically interacting with its environment. State-of-the-art work has accomplished impressive results in dynamic object-tracking with visual input [30, 57, 113]. However, visual techniques are known to be sensitive to occlusions, specularities, transparencies, and poor lighting conditions. This work serves to complement visual methods with audio perception, which is robust to all these challenges. Furthermore, sound intrinsically is a lower-dimensional and higher-frame-rate signal, which consequently makes it much more computationally suitable for real-time robotic systems.

A canonical dynamical system that is of interest to both the physics and robotics communities is the bouncing ball. Indeed, a wealth of prior work in the physics community has utilized impact sounds to estimate a ball's coefficient of restitution, a parameter that abstracts the energy dissipation upon collision [14, 189, 63]. Nevertheless, it is widely acknowledged that modeling a ball's dynamics with a constant estimate of restitution does not realistically capture its stochastic behavior while interacting with the environment. Notably, [63] remarks that "even tiny deviations of the shape from the perfect sphere may lead to substantial errors," specifically, due to directional perturbations upon collision (e.g., a tennis ball dropped on asphalt will likely veer off to a side). Prior work has not addressed this stochastic behavior, yet there is evidently a need for modeling these perturbations.

In robotics, simulations are often used to train robotic tasks with the assumption that the simulator models realistic dynamics, yet simulations are inherently deterministic. However, to capture realistic stochasticity within simulation, three challenges come to mind: 1) How does one automatically infer these stochastic parameters? 2) How can they be incorporated into a simulator? and 3) How can this now-stochastic simulator be utilized in a practical way for real robotic tasks?

Contributions: As more research within robotics seeks to bridge the gap between simulation

and reality, this work contributes to the community in three specific ways (Figure 4.1). First, we present a methodology to automatically calibrate simulators from simple features of sound. Second, we introduce learned stochasticity into a robotics simulator, in which a physics-inspired noise model accounts for real-world distributions observed in a bouncing ball's dynamics, e.g., due to uneven or rough surfaces of the ground or ball. Finally, we conduct experiments to demonstrate the predictability of the calibrated simulator in new scenarios, as well as the use of auditory feedback to achieve reactive control in a robotic ball-tracking task. This work combines techniques from multiple fields so that the resulting contributions may enhance the perceptual intelligence of robots using sound.

4.2 Related Work

Auditory Perception for Robotic Manipulation

Despite its ubiquity and low cost of capture and processing, auditory perception is not often utilized in real-time robotic manipulation. Instead, it has been most widely used for sound localization [146, 103] and object or material classification [18, 28, 95, 93, 190, 185, 23]. However, only few works have successfully employed it for manipulation tasks. [54] showed that sound can be informative of actions applied to sounding objects and thus can be used to predict forward models of the objects. [115] and [31] also demonstrate dynamic inference from sound by using it for feedback control while pouring liquids and grains, respectively. In contrast, in this work, sound is additionally used to learn physically-relevant parameters, which then, combined with sound as feedback, inform fast robotic interactions with a bouncing ball.

Simulation Calibration of Dynamics Parameters

A growing field in robotics lies in estimating the simulation-to-reality gap. Domain randomization is a popular technique used to learn policies in simulation that are transferable to the real world [163, 22]. [169] develops BayesSim, a likelihood-free Bayesian approach to generate mixture-of-Gaussian posteriors over simulation parameters that are used for domain randomization. In general, BayesSim is more sample-efficient and generates better approximations of the posteriors than other likelihood-free inference techniques such as approximate Bayesian computation methods [162]. BayesSim has been successfully applied in different robotic contexts – for instance, in the prior chapter, where BayesSim was used to calibrate granular material simulations, which were then used to predict real-world behavior [133].

This chapter leverages BayesSim to approximate a posterior over a material and noise parameter, which is sampled for domain randomization when simulating a bouncing ball. Both [1] and [4] use real2sim methods on video input to infer bouncing ball parameters. Our work is unique in that sound is used in lieu of vision, and a noise model is learned to account for observed stochasticity. Learning stochastic parameters has been shown to be useful in works such as [10, 9, 123], where probabilistic models are used to make predictions of planar pushing with frictional contact.

The only known works other than this chapter that use sound for real2sim focus on realistic sound synthesis [229]. However, [229] acknowledge their limitations in sim2real transfer, as they rely on bridging the gap using high-dimensional spectral features of sound. In contrast, this chapter extracts simple features from audio, making it robust to discrepancies in vibration representations in simulation and real life as well as more suitable for real-time applications.

Robotic Bouncing Ball Tracking

A bouncing ball is difficult to track due to uncertainties in collision dynamics. However, robot interactions with a dynamic object require accurate tracking and trajectory predictions. Several works have attempted bouncing-ball tracking with vision, either by using high-speed or multiple cameras [57] or pairing low-cost cameras with sim2real tuning of physically-relevant parameters [1, 4]. By using dynamics models trained in the tuned simulator, trajectory predictions compensate for tracking uncertainties [4]. In contrast, this work presents ball tracking through sound and physical parameter inference. The only known work that uses sound to interact with a bouncing ball is [99], where the author designed a mechanism that continuously bounced a ball off an actuated platform. However, sound is used reactively, rather than to inform predictions about the ball dynamics.

4.3 Methods

The STReSSD framework combines BayesSim [169] with audio processing to calibrate a stochastic dynamic simulator. We highly recommend referring to Section B, which illustrates an expanded version of the STReSSD framework. The following sections detail each module in the framework.

Simulation Calibration with BayesSim

This work aims to calibrate a physics simulator using auditory signals of ball-table collisions to closely predict real-world dynamics of the ball. Let θ be the simulation parameters to be inferred, and let X represent information-rich features of an auditory signal. Assuming that there is an equivalent representation of X in simulation (X^s) and the real physical environment (X^r), then the problem can be formulated as computing the posterior $p(\theta|X^r)$ over the simulation parameters. As in Chapter 3, this work uses a likelihood-free framework called BayesSim [169] to approximate this posterior function. Specifically, N pairs of parameters and feature vectors $\{\theta_i, X_i^s\}_{i=1}^N$ are first generated from forward simulations of samples from a physically-motivated prior distribution, $p(\theta)$. A conditional density function $q_\Phi(\theta|X^s)$ is then learned by mapping extracted simulation features X^s to a mixture of K Gaussian components with a mixture density neural network (MDNN), parameterized by Φ. The conditional density function q_Φ, along with a real observation X^r, are used to generate an approximation of the posterior $p(\theta|X^r)$. The main advantage of using BayesSim over non-Bayesian techniques (e.g., classical optimization) is that the approximated

posterior can be informative of the uncertainties regarding the parameters θ, as well as sensitivity to measurement noise.

Stochastic Simulation of a Bouncing Ball

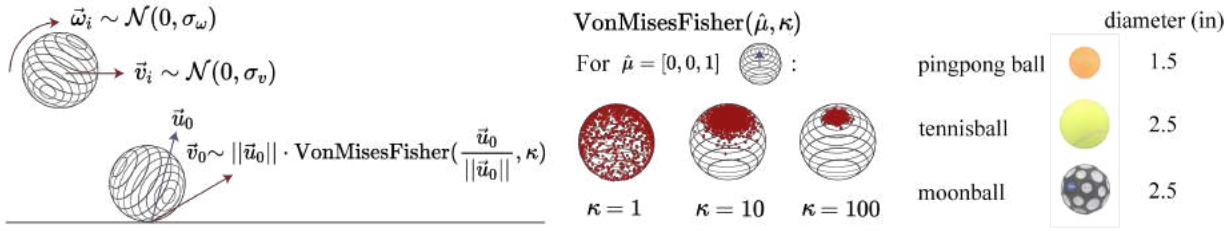


Figure 4.2: (Left): The initial velocity is sampled from a Gaussian distribution and collision perturbations are sampled from a von Mises-Fisher distribution, centered at $\vec{u}_0/||\vec{u}_0||$. (Middle): 1000 samples over a sphere for different values of κ for $\mu = [0, 0, 1]$. (Right): Balls used in real experiments.

Here, we define the simulation parameters θ that are used to model the stochastic dynamics of a bouncing ball. The coefficient of restitution e is a standard ratio used to quantify energy loss upon object collision. For an object bouncing on an immovable surface, e can be calculated as the ratio of the exiting to entering velocity of the collision. However, as velocities are difficult to extract from sound, bounce times are used to calculate $e = \frac{\text{time from bounce } i \text{ to } i+1}{\text{time from bounce } i-1 \text{ to } i}$ (see Section B.1).

While e is generally used as a *constant* in simulation, numerous factors contribute to nontrivial variance in measurements of e [141]. For example, if a ball is dropped vertically onto a level flat surface, surface asperities cause the ball to experience horizontal perturbations. Because we are measuring e from bounce times instead of velocity, these horizontal perturbations produce a *distribution* over e.

[141] address this variance by fitting a probability density over e. Instead, this chapter aims to represent e and surface asperities as independent variables. To simulate collision perturbations, [141] represent a ball as a composite multisphere particle with $\sim 10^6$ surface asperities. In contrast, this chapter reduces computational complexity by imitating the stochastic dynamics of a bouncing ball with a simple noise model. At every collision, we apply a random perturbation, sampled from a Gaussian distribution on a sphere (i.e., the von Mises-Fisher distribution), to the direction of the outgoing velocity vector. The von Mises–Fisher distribution on a 2-sphere (Figure 4.2) is given by the probability density $f(x; \mu, \kappa) = \frac{\kappa}{2\pi(e^\kappa - e^{-\kappa})} e^{(\kappa \mu^T x)}$ for 3-dimensional random unit vector x, where μ is the mean direction and κ is a concentration parameter.

Let $\vec{u}_0$ represent a ball's unperturbed exit velocity during a collision. To simulate a collision perturbation (e.g., due to rough surfaces), a vector is sampled from $f(x; \hat{\mu}, \kappa)$, where $\hat{\mu} = \frac{\vec{u}_0}{||\vec{u}_0||}$. The vector is then scaled by $||\vec{u}_0||$ to produce a perturbed exit velocity $\vec{v}_0$. Thus, the outgoing

kinetic energy of the ball is unaffected by the perturbation. Stochastic ball dynamics are therefore modeled by e, for energy dissipation, and κ, to account for the stochasticity due to collisions.

While this model can be integrated with most standard simulators, we used NVIDIA's Isaac Simulator paired with a preconditioned conjugate residual (PCR) solver [125]. This GPU-optimized simulation platform enabled the parallelization of hundreds of environments. This was particularly useful for efficiently generating distributions of ball trajectories due to the modeled collision perturbations.

Audio Processing

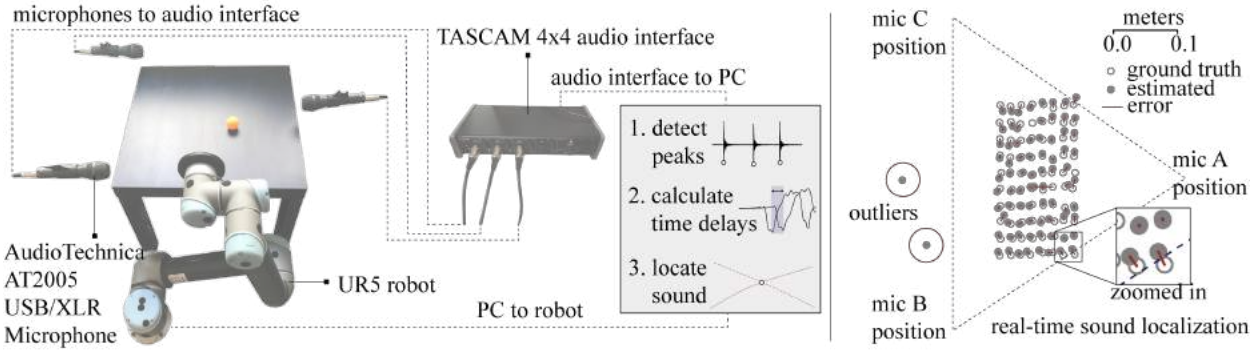


Figure 4.3: (Left): Integrated robotic system used in experiments. (Right): Real-time sound localization is slightly less accurate than the offline method and produces outliers with large errors.

We hypothesized that the times and positions of the ball bounces are informative of $\theta = \{e, \kappa\}$, as e can be expressed as a function of impact times (Section 4.3), and κ affects the ball bounce positions. This motivates our four-dimensional feature vector X, which encodes both temporal and positional information. Letting t_i, $\vec{p}_i$, and d_i denote the time, positional vector, and distance between bounce i and $i+1$, then X is composed of $\frac{t_2}{t_1}$, $d_1 = ||\vec{p}_1||_2$, $d_2 = ||\vec{p}_2||_2$, and α, the signed angle between $\vec{p}_1$ and $\vec{p}_2$ (see Section B.1). Higher dimensional spectral features were not needed to estimate θ, but future work could leverage this additional information to, for instance, classify the ball's material.

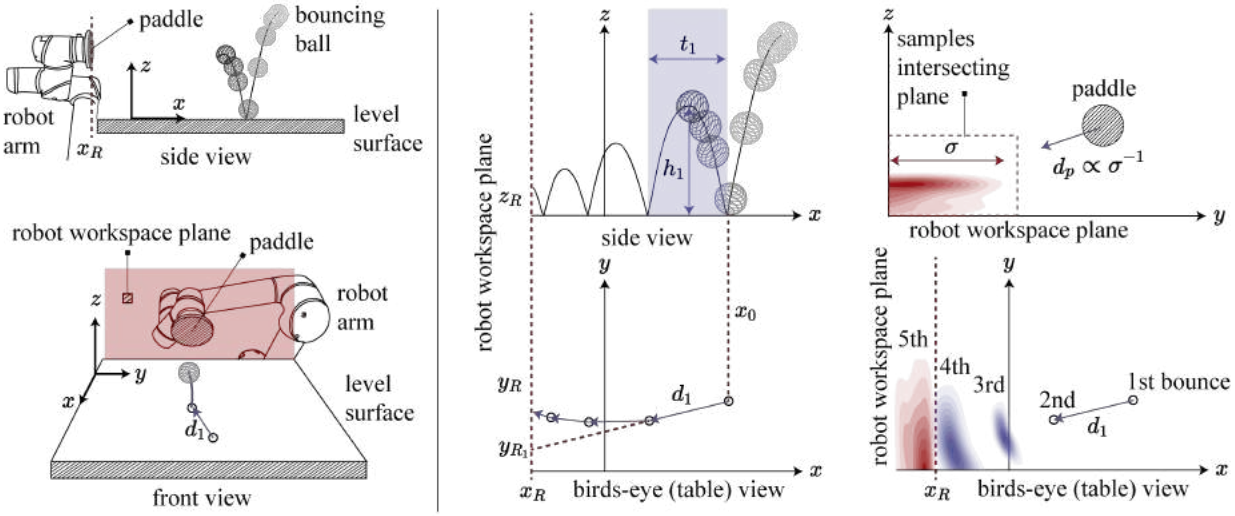
Vector X is simple to extract from simulation, but, in the real-world, depends on accurate sound localization for each bounce. We used two localization methods – an offline method for parameter inference and an online method for real-time interaction, both of which relied on the same principle of time delay estimation (TDE). TDE uses time delays between microphone signals to locate sound sources (see Section B.2 for derivation). For 2D localization, only 3 microphones are necessary.

The offline and real-time methods diverged in how these time delays were calculated. Offline time delays were calculated using phase correlation, a standard frequency-domain technique [98], over audio segments of $>$20 ms. In contrast, real-time processing required fast calculations on 11 ms-length buffers of incoming audio. Peak times corresponding to the strong peaks induced by

collisions could be detected efficiently and thus were used to calculate time delays. The offline method was found to be slightly more accurate, with an average error of about 6.7 mm compared to real-time's 7.9 mm, shown in Figure 4.3 (see Section B.2 for comparisons). Experiments were conducted in a household environment rather than a sound studio, so real-time localization was slightly less robust. The method's sensitivity to reverberation, or echoes from early reflections off of walls and furniture, occasionally caused large errors in detecting true impact times. However, the performance was a necessary trade-off to enable reactive robot interactions. Furthermore, peak detection was generally unaffected by other factors such as noisy robot motion or acoustic differences in ball material.

For experiments, auditory signals were captured by three AudioTechnica AT2005 USB / XLR Microphones, which were placed around a 0.55-by-0.55 meter particle-board table, 3 inches above the table surface. Microphone signals were streamed at 44.1 kHz in parallel to a PC through a TASCAM 4x4 audio interface (Figure 4.3). For robot experiments, audio was processed in real-time, and control signals were sent to a UR5 robot in parallel processes. The overall latency of the system was dominated by robot motion, as bounce localization and inference ran at $\sim$100 and 20 Hz, respectively.

Robotic Ball Tracking



(a) experimental setup for ball-tracking task (b) dynamic predictions without noise (c) dynamic predictions with noise

Figure 4.4: (a) Ball tracking setup. (b) Example ball trajectory for noiseless method. y_{R_1} is the y-coordinate of the inferred crossing point of the ball given the first two bounces. (c) Samples of future bounce positions given the first two bounces using the stochastic noise method. The mean and variance of the samples in the workspace plane inform the distance the paddle should travel (d_p).

To illustrate the benefits of modeling dynamic stochasticity and the real-time capabilities of audio perception, we compare two methods to accomplish the robotic task of tracking and deflecting a bouncing ball using sound. The methods differ in their consideration of perturbations upon collision. Pseudocode and equations accompanying each method are presented in appendices B.3 and B.4.

Deterministic Baseline: Dynamic Predictions without Collision Noise

Without modeling collision perturbations (thus ignoring concentration parameter κ), the dynamics of a bouncing ball are defined only by its initial conditions, gravity, and energy-dissipating forces. For this particular method, we assume that the most significant parameter affecting ball dynamics is e. The ball trajectory can then be defined by classical projectile equations, with e applied to simulate energy dissipation at each bounce. Figure 4.4a-b illustrate the control scheme used to guide the robot to contact the ball as it enters the workspace plane. The ball's future bounces can be extrapolated from the times and positions of the last two bounces, since here we assume that the ball is not directionally perturbed. The goal position of the robot, denoted as (x_R, y_R, z_R) in Figure 4.4, is then defined as the intersection of the piecewise parabolic trajectories with the robot plane. If the ball will intersect the plane at t_R seconds before the next bounce, then the robot moves to the goal position in that time. Otherwise, the robot moves to y_R before the next bounce is observed.

Stochastic Method: Dynamic Predictions with Collision Noise

Predicting the dynamics of a ball without incorporating knowledge of collision noise may cause the robot to be over-reactive to small perturbations in the ball's trajectory. Furthermore, success relies on accurate real-time sound localization, which as mentioned earlier, can occasionally be sensitive to reverberations, causing large errors in the inferred bounce location (Figure 4.3). Thus, we aim to incorporate the noise model to address two goals: leveraging the uncertainty of the ball's location to prevent the robot from over-reacting, and filtering unrealistic inferred bounce locations.

Both goals employ a second BayesSim model that generates posteriors over the next bounce time and location, given the time and distance between the prior two bounces. The BayesSim model, trained via the calibrated simulator, serves as a probabilistic dynamics model. At run-time, a generated posterior is used with an elliptic envelope filter to classify if the observed bounce can be trusted (or if it is an outlier). If the latter, the mean of the posterior is used to estimate the bounce location.

In addition, the ball's future trajectory is estimated by generating samples from the posterior, which in turn are used to generate new posteriors. Thus, trajectory predictions are made by recursively generating posteriors until samples intersect the plane. In this new context, BayesSim is used as a generic regression technique to learn a transition model, which is applied recursively, and whose predictions are averaged. This dynamics model could also be incorporated into a nonlinear filtering technique such as an Unscented Kalman Filter, although this would require defining a sensor model. Figure 4.4c shows that the predicted future bounce location uncertainty grows with every bounce. This uncertainty informs the end-effector motion. The smaller the variance of

the predictions are, the closer the end-effector will move toward the goal location before the next observation.

4.4 Experiments

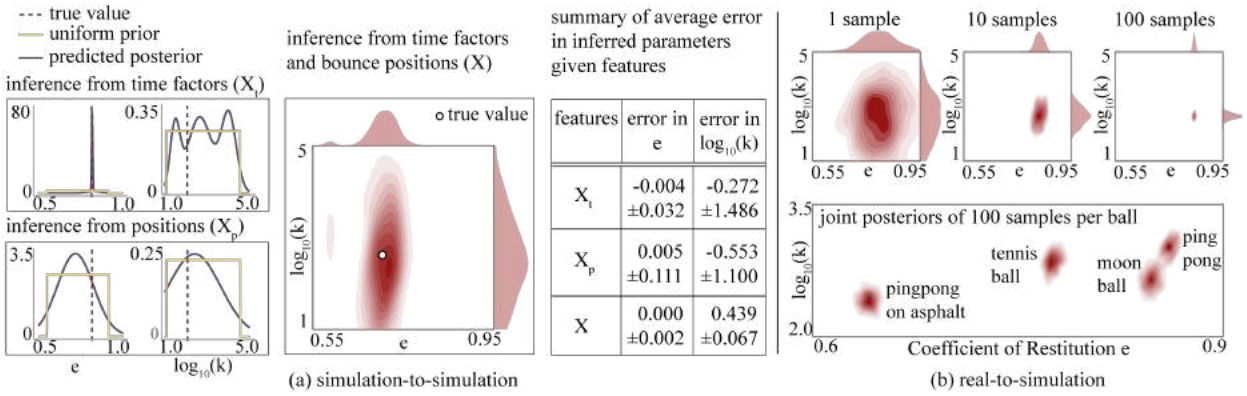


Figure 4.5: (a) Sim-to-sim example posteriors of inferring e and $\log_{10}(\kappa)$ from bounce times, e and $\log_{10}(\kappa)$ from bounce positions, and both e and $\log_{10}(\kappa)$ from times and positions. The table summarizes average error over 1000 examples for each parameter, given a feature subset. (b) Top: Real-to-sim posteriors become more certain with more samples. Bottom: Posteriors for real balls.

Simulation-to-Simulation Experiments

The inference framework was first tested with simulated data to assess whether bounce locations and times were relevant measurements for the inference of the desired model variables (e and κ). Training data was generated in simulation by sampling from a physically-motivated uniform prior distribution, with $e \in [0.55, 0.95]$ and $\log_{10}(\kappa) \in [1, 5]$. Each simulation sampled an initial angular and linear velocity from a Gaussian centered at 0 (see Figure 4.2) to mimic real-world noise when dropping a ball. 6000 and 1000 simulations were run for the training and testing dataset, respectively.

To measure how informative each feature in X was for each parameter in θ, an ablation study was performed over the simulated dataset. The features were divided into two sets, one relevant to time ($X_t = \frac{t_2}{t_1}$), and the other, to position ($X_p = \{d_1, d_2, \alpha\}$). Four BayesSim models were trained, each using either X_t or X_p to infer one of e or κ. These models were used to generate posteriors from a testing dataset of 1000 examples, and the peaks of the posteriors were compared with the true value of the inferred parameter. The table in Figure 4.5a summarizes the average error in inferred parameters e and κ, given a subset of features. Posteriors that are unimodal and low-variance have higher certainty and thus relevance to the inferred parameter. As reflected in the single-parameter posteriors of Figure 4.5a, time appears to be a more informative observation for e, while

positional information is more relevant for κ. Knowledge of both time and positional information generates significantly more certain posteriors. Observe that the variance of the marginal posterior for $\log_{10}(\kappa)$ will naturally be greater than that of e, as estimating a noise parameter is inherently more difficult.

Real-to-Simulation Experiments

With the assurance that X contained informative features for θ, we turned to extracting θ to describe real bouncing balls. Note that both e and κ are reflective of material-to-material interactions, and thus are unique to each pair of object and surface. The four object-surface pairs tested were a ping-pong ball, tennis ball, and moon ball (which has large crater-like divots) bouncing on a smooth table, and a ping-pong ball bouncing on an asphalt surface (see Figure 4.2). This was to demonstrate the effect of surface asperities both on the ground (asphalt) as well as on the ball (e.g., moon ball).

For each ball-surface pair, the ball was dropped 100 times from 0.26 meters (to match simulation), and offline sound localization captured the locations and times of the bounces. Real-world feature vector X^r was extracted for each drop and used to generate a posterior over θ. Figure 4.5b shows an example posterior generated from one sample, or drop, of the moon ball. Note that, from one sample, the posterior is quite uncertain. However, assuming independence between each new observation X_i^r, the joint posterior can be generated by multiplying each individual posterior together. As the single sample posteriors reflected strongly Gaussian structures, each mixture-of-Gaussians was projected to a single Gaussian before multiplication. As shown in Figure 4.5b, the joint posterior became more certain with more observations, with the joint posterior of 100 samples demonstrating strongly peaked marginals in both e and $\log_{10}(\kappa)$. The resulting posteriors of each ball-surface pair are illustrated in Figure 4.5b, and their positions relative to each other qualitatively reflect their real-world dynamic characteristics (e.g., the moon ball is bouncier and noisier than the tennis ball).

Closing the Loop: Real-to-Sim-to-Real

In order to test if the dynamic model of the bouncing ball generalized well to new scenarios, an experimental setup similar to the one constructed in [1] was used, as illustrated in Figure 4.6. The table surface used in Section 4.4 was tilted at an angle of 10.72 deg, and a plastic cup was placed 8.6 inches away from the bottom of the incline. The physical experimental setup was replicated in simulation, and the goal was to compare the success rate of bouncing a ball against the inclined plane into the cup in simulation and in real experiments. While a deterministic simulator would only either fail or succeed if a ball was dropped at a specific initial position, by modeling dynamic noise due to surface perturbations, our stochastic simulator had probabilistic success.

Using a Robotiq 2F-140 gripper to drop the ball repeatedly over the inclined surface at a fixed position, real experiments showed similar probabilistic levels of success, with the pingpong ball and moon ball bouncing into the cup (out of 100) 75 and 35 times, respectively. The experiment was repeated in simulation 30 times, for each ball and their respective drop positions. For each

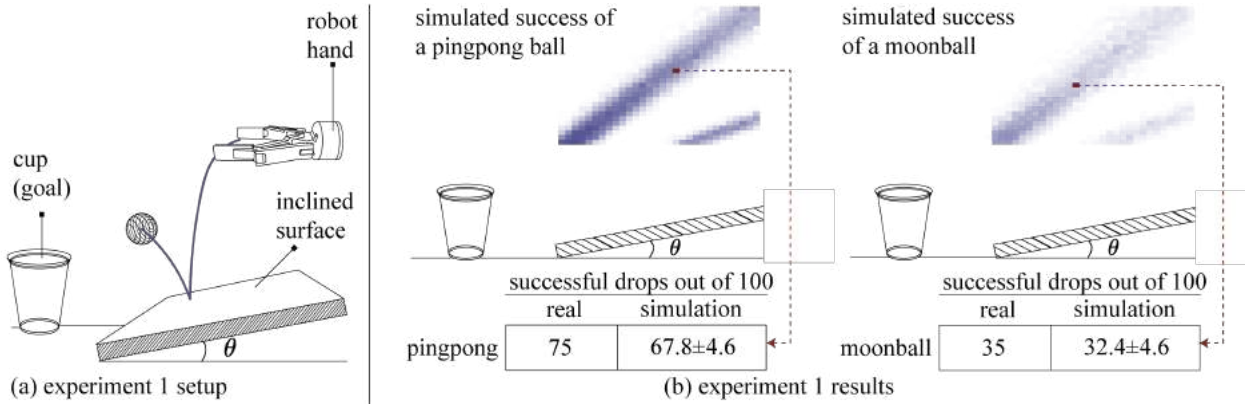


Figure 4.6: (a) A robotic hand bounces the ball off the tilted table surface into the cup, or goal. (b) Top: Spatial map of success rates for different initial positions of the ball drop. Success ranges from lightest (0 successful) to darkest (100 successful). Bottom: Comparison of real and simulated success rates (out of 100) dropped at the red point, with simulated success rates averaged over 30 trials of 100 simulations.

simulation, a new pair of e and $\log_{10}(\kappa)$ were sampled from the inferred posteriors from Section 4.4. The average simulated success rates were within close range of the real success rates, suggesting that the learned dynamics model could generalize well to novel physical scenarios. This also implies that the simulator could potentially be used for predictive measures, e.g., searching for where to drop a ball to maximize success rate. To demonstrate this concept, we sampled success rate (out of 100) over a discrete number of initial positions in simulation. This generated a spatial map of probabilistic success (Figure 4.6), a capability made possible through the modeling of collision perturbations.

Robotic Ball-Tracking Results

The goal of the ball-tracking task was to move the robot end-effector, a circular paddle with a diameter of 4.5 inches, to touch the bouncing ball as it entered the robot workspace plane, using only sound as feedback. Each of the methods described in Section 4.3 was evaluated for the tennis ball, ping-pong ball, and moon ball on the following metrics: 1) success out of 30 trials (whether or not the end-effector touched the ball), and 2) kinetic energy of the end-effector, integrated over the trajectory. We hypothesized that, by modeling collision perturbations, the probabilistic method would be more robust to errors in bounce localization as well as curb the reactiveness of the robot to small perturbations in the ball's trajectory. Both methods assumed nothing about the initial condition of the ball, and for each trial, the ball was manually tossed onto the table in the direction of the robot.

As shown in Figure 4.7, the stochastic method averaged an absolute increase in success rate of 24%, which corresponds to an improvement of 70%, relative to the deterministic baseline. Failure

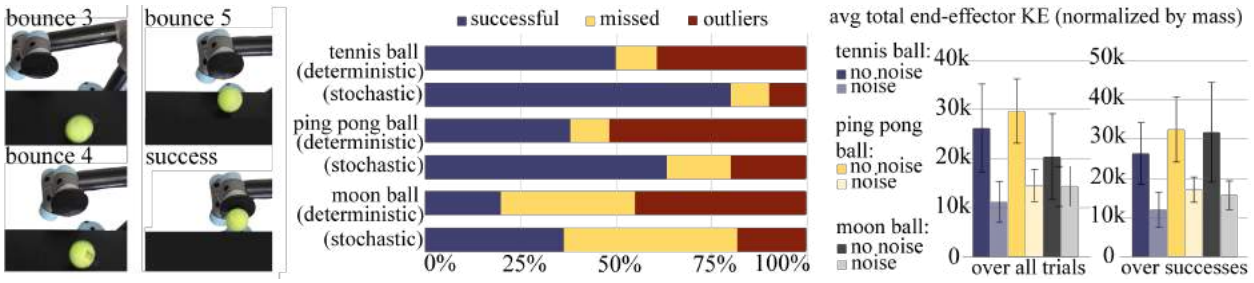


Figure 4.7: (Left): A successful trial. (Middle) "Outliers" signifies failures due to sound localization errors, and "Missed" denotes any other failure. Compared to the deterministic baseline, the stochastic method has higher success (an absolute increase of 24%) and fewer failures due to outliers (an absolute decrease of 29%), averaged across all balls. This graph excludes trials that violated the robot's boundary constraints; an alternative graph with all 30 trials can be found in Section 15. (Right): On average, the baseline uses more total energy than the stochastic method over all trials and all successful trials, suggesting that the baseline is more overreactive.

cases are labeled as "outliers" for errors in bounce localization and "missed" for all other failures. For each of the balls, a few of the 30 tosses resulted in the ball violating the robot workspace boundaries, so those tosses were removed from analysis. Sensitivity to "outliers" in bounce localization can be resolved by using dynamic predictions with collision noise described in Section 4.3. Figure 4.7 precisely demonstrates this effect, where the number of failure cases due to outliers in bounce localization diminished for all three balls. While the stochastic method eliminates failures due to outliers in later bounces, it is still sensitive to outliers in the first and second bounces, since it relies on those to generate posteriors for future bounces. Finally, Figure 4.7 shows that the average total end-effector kinetic energy over all trials, normalized by mass and averaged across balls, was reduced by 46% when considering collision noise versus without. This pattern is replicated when looking at the average energy over just the successful trials (a reduction by 51%), suggesting that the stochastic method does indeed curb over-reactions to small perturbations in a bouncing ball's trajectory.

4.5 Conclusions

This work presents STReSSD, a novel framework for calibrating a simulator for stochastic dynamics from simple, low-dimensional features of sound. A physically-motivated noise model is learned from real audio, which enables a physics simulator to reflect real stochastic dynamics caused by rough or uneven surfaces. Using the canonical dynamical system of a bouncing ball, this work closes the loop on real-to-sim-to-real by demonstrating the predictive capabilities of the learned dynamics model in both an open-loop and sound-in-the-loop robotics task. To the best of our knowledge, this is the first robotics work that relies solely on sound to accomplish a highly dynamic, probabilistic task. We believe that the robotic capabilities demonstrated by the devised

framework highlight the utility of audio perception and may encourage increased integration of sound in robotics.

While the experimental results are promising, there are various opportunities to extend this work. Sensor fusion with vision and more robust peak-detection can improve the accuracy of real-time sound localization. Furthermore, one could imagine utilizing the rich spectral information from the raw audio signals to classify the material of the ball. The multivariate classification uncertainty could then be incorporated into the BayesSim model to produce a multimodal posterior distribution over simulation parameters. Other paths for future work include exploring simulation-in-the-loop inference, where the simulator is calibrated in real-time and a dynamic object is tracked using a particle filter or an Unscented Kalman Filter. Finally, we believe the framework in this chapter may be revised to perform granular material parameter estimation using audio signals of grain collisions. These advancements may lead to broader impact in various sub-fields of robotics, including navigation and unseen-object manipulation.

Part III

Doughs and Deformable Objects: Designs of Variable Stiffness Tactile Devices

For the remainder of this thesis, we explore the concept of embedding interactive perception into the design of a tactile sensor. Chapters 5, 6, and 7 present different iterations of variable stiffness soft tactile sensors that serve also as actuators. Through the evolution of this concept, this technology ultimately enables the robot to interactively perceive and manipulate deformable objects like dough.

Each soft tactile sensor presented in Chapters 5-7 can be categorized as a *soft optical tactile sensor*. The tactile interface is composed of *soft*, compliant material, and the deformations of the membrane's surface, resulting from contact with the environment or an object, are imaged by an *optical* sensor (e.g., an RGB-D camera). Development of this category of tactile sensors has seen increased interest in the field of robotics. Much of this aligns with ambitions of expanding the capabilities of dexterous manipulation. For instance, the GelSight sensor [221] captures high resolution contact images, thereby enabling tasks such as USB orientation estimation [114] with touch. Furthermore, soft tactile sensors such as the TacTip [206] and Soft-Bubble [2] can safely conform to surrounding geometries due to their high compliance, enabling a robot to gently palpate tissue for tumor detection or robustly handle delicate objects like a wine glass. In [224], the authors use a GelSight sensor to estimate the hardness of deformable objects, noting that the GelSight performs best when the "hardness" of the object is closest to the hardness of the membrane. This observation motivates each device presented in Part III, where they are designed to vary in stiffness in order to expand both sensing and actuation capabilities for robotic manipulation.

We begin by studying the sensing-actuation duality of a pneumatically-actuated variable stiffness tactile sensor in Chapter 5. Chapter 6 then modifies the design of the sensor in Chapter 5, making it simpler to manufacture and integrate with an industrial robotic hand. This new design iteration, called StRETcH, is used to measure both the geometry and a proxy for stiffness of deformable objects like a water balloon. Part III concludes with Chapter 7, where a modified version of StRETcH is used to roll a ball of dough into logs of varying lengths. Proxy estimations of stiffness, measured through active palpation, are shown to enable more efficient dough rolling.

Chapter 5

A Soft Fingertip with Integrated Sensing and Actuation

This chapter is based on the paper "Towards a Soft Fingertip with Integrated Sensing and Actuation" [136], written in collaboration with Benjamin McInroe, Ken Goldberg, Ruzena Bajcsy, and Ronald Fearing.

Soft material robots are attractive for safe interaction with humans and unstructured environments due to their compliance and low intrinsic stiffness and mass. These properties enable new capabilities such as the ability to conform to environmental geometry for tactile sensing and to undergo large shape changes for actuation. Due to the complex coupling between sensing and actuation in high-dimensional nonlinear soft systems, prior work in soft robotics has primarily focused on either sensing or actuation. This chapter presents SOFTcell, a novel controllable stiffness tactile device that incorporates both optical sensing and pneumatic actuation. We report details on the device's design and implementation and analyze results from characterization experiments on sensitivity and performance, which show that SOFTcell can controllably increase its effective modulus from 4.4kPa to 46.1kPa. Additionally, we demonstrate the utility of SOFTcell for grasping in a reactive control task in which tactile data is used to detect fingertip shear as a grasped object slips, and cell pressurization is used to prevent the slip without the need to adjust fingertip position.

5.1 Introduction

There is a growing interest in using soft materials, such as elastomers and polymers, in robots. Two widely recognized advantages of such systems are the ability to deform and change shape in response to internal and external loads, and their low intrinsic stiffness and mass, which makes them especially useful for safe human-robot interaction and operation in delicate environments. Previous studies have explored the capabilities of soft systems in both actuation and sensing [175, 205], and recent progress in soft robots has shown them to be capable of performing challenging

tasks, including legged locomotion [201] and dexterous grasping [76].

Design of actuation, sensing, and control of these soft systems remains an active area of research. Common methods of actuation include pneumatic [142] and hydraulic [166] actuation, cable-driven tendon systems [106], smart materials such as shape-memory alloys (SMAs) [204] and dielectric elastomers (DEAs) [83], and chemical reactions [207]. The high dimensional configuration spaces and intrinsic nonlinear dynamics of soft materials present both a challenge for control and an opportunity for extracting rich tactile information from the external environment. Many tactile sensing technologies have been integrated into soft robotic platforms, including fiber optic sensors [105] and flexible electronic sensors [120].

However, the embedding of rigid materials such as conductive wires or tendons in soft components imposes constraints on maximum material strain [48]. This limits the deformation ability of the system, and thus removes a fundamental benefit of soft material robots. To circumvent this limitation of traditional tactile sensors, non-contact sensors for soft systems have been explored [42] in order to decouple the sensing devices from the soft material. For pneumatically-driven systems, one solution that has been explored is using pressure sensing to detect collisions [191]. Image-based tactile sensors, which measure 2-D and 3-D deformation fields from image data, present another attractive alternative. Spatially rich tactile information enables contact localization and registration of finer contact dynamics.

Non-contact image-based methods have been previously used to develop soft fingertips for tactile sensing and dexterous manipulation [172]. Begej et. al. [11] designed planar tactile sensors that used cameras to image contact on an elastic membrane. Yamaguchi et. al. [211] developed a low-cost tactile sensor called FingerVision that uses a transparent skin with tracking markers to collect static and dynamic information. Hristu et. al. [69] built a gel-filled fingertip with markers and a fixed camera that measured displacements of markers on the fingertip surface to reconstruct 3-D geometries. A related device is the GelSight sensor, a high resolution tactile sensor with an elastomer gel membrane coated with reflective material, which uses a coaxial camera and illumination by multiple LEDs to capture height maps of the surface deformation [114]. Chorley et. al. [27] developed a similar device to that of [69] called the TacTIP sensor, which includes pins on the surface of the gel-filled membrane inspired by dermal papillae in human fingertips. The WORMtip sensor [66] replaced the gel-filled membrane with a pneumostatic vessel consisting of a dielectric elastomer (DEA) sheet on one end and an elastic membrane on the other. Actuation of the DEA allowed the surface to extend and retract for palpation tasks. However, kilovolt ranges are needed for DEAs in comparison to pnuematically actuated robots that can operate through miniaturized compressors and valves that typically run at household voltages [201].

Recent work has explored the possibility of function-selectable tactile sensing through programmable morphology [213]. However, the potential for both multimodal sensing and actuation in soft tactile sensing devices remains underexplored. Stilli et. al. begins to bridge this gap by proposing to use pressure sensing of pneumatically-driven robotic arms to detect collisions [191]. More recently, Huh et. al. present a novel soft, suction-based tactile sensor that directly measures surface conditions using active suction flow [74]. We aim to continue to reveal the capabilities of soft actuated tactile devices through the development and study of a new pneumatically-actuated soft tactile sensing device, the SOFTcell (Soft Optical Feedback Tactile cell). Through the combi-

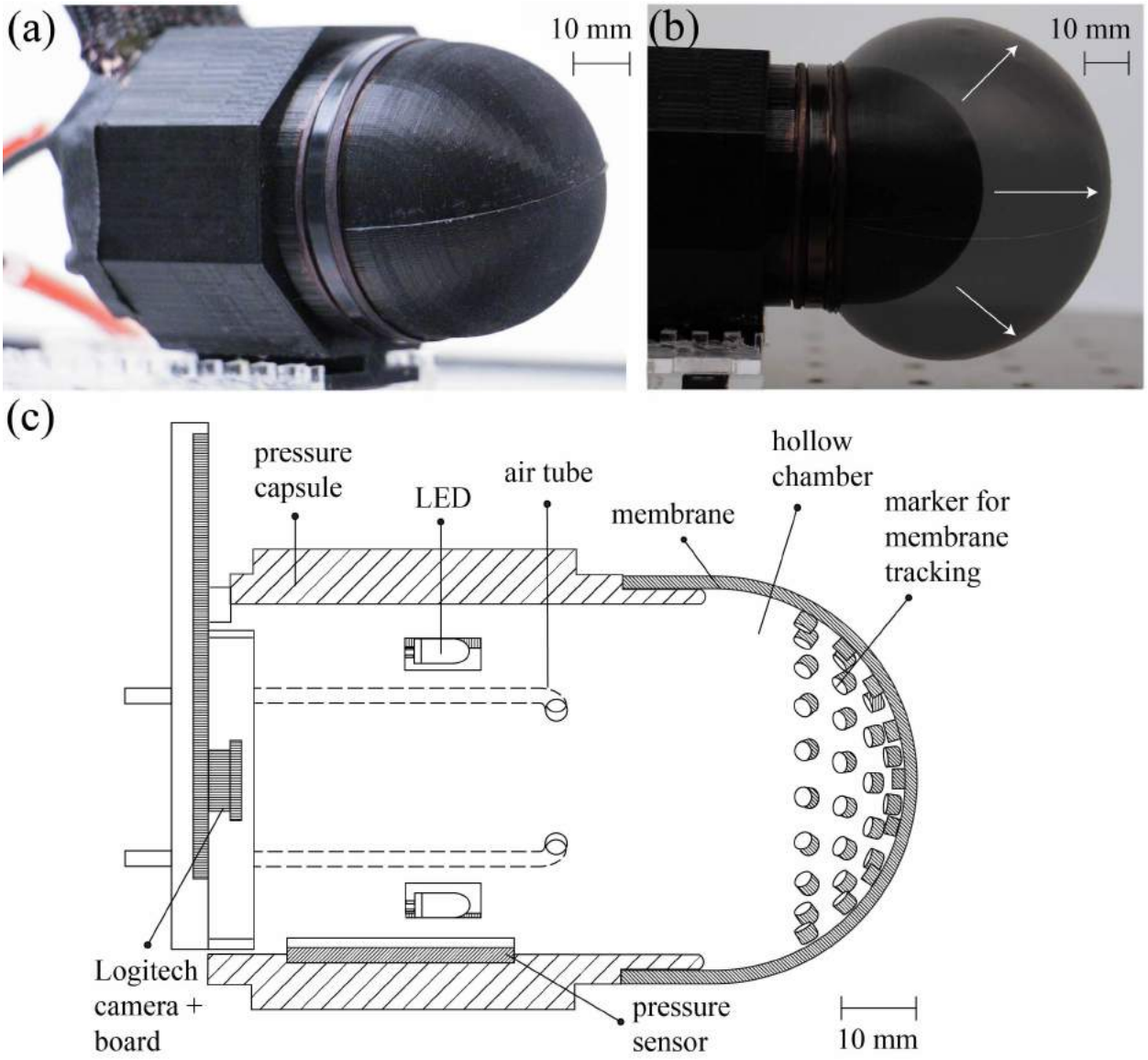


Figure 5.1: (a) The Soft Optical Feedback Tactile cell (SOFTcell), a device towards a soft fingertip with integrated sensing and actuation. (b) Pneumatic actuation enables controllable increase of effective modulus and a greater than 200% increase in volume. (c) A cutaway scale diagram of SOFTcell displaying its assembled internal components. © 2018 IEEE

nation of valve-controlled pneumatic actuation and image-based sensing, we controllably exploit the intrinsic conformability of a soft membrane at low internal pressure for large deformation tactile sensing and the increased effective modulus at high internal pressure for large load actuation. We show that the functionality of the system is multimodal, with different pressurization states allowing for different functionality such as improved tactile sensitivity or increased contact forces.

We further explore the utility of SOFTcell's multimodal sensing and actuation in practical

applications by demonstrating shear detection and grasp adjustment in a constrained manipulation task. SOFTcell holds an object against a static wall, and the mass of the object is slowly increased. When shear is detected due to the object slipping against the wall, the cell pressure is increased, stabilizing the object without the need to adjust fingertip position.

The chapter is organized as follows. Section II discusses the design and fabrication of SOFTcell, including a discussion of the general work flow and image processing algorithms used for tactile feedback. Section III presents a theoretical analysis of SOFTcell in the linear elastic regime and describes the fundamental physical principles governing the controllable sensing and mechanical stiffness capabilities of the device. Section IV presents the results of systematic experimental characterization of SOFTcell and discusses empirical measurements of variable stiffness and tactile sensitivity. Section V presents empirical measurements for model validation and describes preliminary application-domain demonstrations in reactive grasping that employ both high resolution sensing and large deformation actuation. Section VI further discusses the utility of SOFTcell and coupled soft sensor/actuator systems, followed by a discussion of future directions.

Summary of Contributions

This chapter makes the following contributions:

1. The design and fabrication of SOFTcell, a controllable stiffness device that is capable of both spatiotemporal tactile sensing and actuation.

2. Modeling and characterization of SOFTcell's controlled stiffness range and multimodal sensing and actuation features.

3. A demonstration of the application of SOFTcell in a reactive control task, which requires both spatiotemporal tactile data and actuation to exert controlled forces on the environment.

5.2 System Design and Fabrication

Design and Fabrication

Tip

The underside of the silicone skin closely resembles that of [27], a pattern of cylinders that are inspired by Meissner's corpuscle mechanoreceptors for sensitive tactile responses. This patterning was chosen primarily due to its area-preserving characteristics, an important feature to simplify image processing for a membrane that is capable of nontrivial changes in surface area. As shown in several relevant works on tactile optical sensors ([211, 50, 223, 66, 33]), tracking features of deformable skin such as these synthetic markers allows for acquisition of tactile data, e.g. dynamic information like shearing or geometric information like convexity of an object. We designed and printed a hemispherical multiple-part mold to cast the tip out of platinum-catalyzed EcoFlex 50 Silicone. In order to ensure uniform thickness throughout the membrane, a vacuum chamber was used to remove gas from the uncured silicone. The cured silicone skin has a thickness of 1.7 mm and an inner radius of 21.23 mm. Each cylinder is 2mm long with a radius of 1mm, and the

cylinders are distributed uniformly across the underside of the hemisphere with a center-to-center distance of at least 7mm. To make the cylinder features visually striking, black silicone pigment was mixed into the EcoFlex and the cylinder tips were painted with white silicone-based paint. The silicone membrane was fastened onto the rim of the camera and pressure capsule with cable ties.

Camera and Pressure Capsule

A Logitech HD C310 Webcam, centered and facing towards the underside of the silicone skin, was secured at the bottom of a capsule 3D-printed out of Polylactic Acid (PLA). The camera was positioned 67 mm away from the rim of the capsule to optimize for its focal length and field of view. A pressure/temperature MS5803-14BA sensor with an accuracy of ± 0.02 kPa under pressures up to 1400 kPa in room temperature was also fastened inside the capsule. Flexible pipes embedded into the capsule's walls act as channels between solenoid valves connected to an air compressor and the capsule's interior, allowing the capsule to be pressurized. The capsule was sealed with hot glue to minimize leakage of air.

Composite Setup

The device consists of the hemispherical silicone membrane attached to the pressure capsule, with LED's for lighting and a camera at the bottom of the cell for sensing. A cutaway diagram of the composite setup of SOFTcell and its internal components is illustrated in Figure 5.1. The entire cost of the SOFTcell, including all sensors and materials, amounted to under 100 USD.

System Configuration

The SOFTcell is connected to a compressor pressurized to 410 kPa, and the solenoid valves that modulate air flow from the compressor into the SOFTcell are controlled by Matrix driver boards. The valves are fitted with orifices that leak air so that pressure can be increased and decreased by a single Pulse Width Modulated (PWM) solenoidal valve [143].

An Arduino Micro microcontroller relays PWM commands to the valve driver boards. The Micro and camera are both connected to a central computer. Application demonstrations reported in Section 5.5, which require basic feedback from the camera images, involve serial communication between a Python script and the Micro.

For the characterization experiments reported in Section 5.4, an ATI 45 six-axis force-torque transducer is oriented in front of the SOFTcell in various configurations in order to measure the force and torque output of the device. A second Arduino microcontroller is used to receive data from the MS5803-14BA pressure/temperature sensor and an ATI F/T Data Acquisition (DAQ) device is used to relay the measurements from the force-torque transducer to a MATLAB script, which we use to further analyze the data.

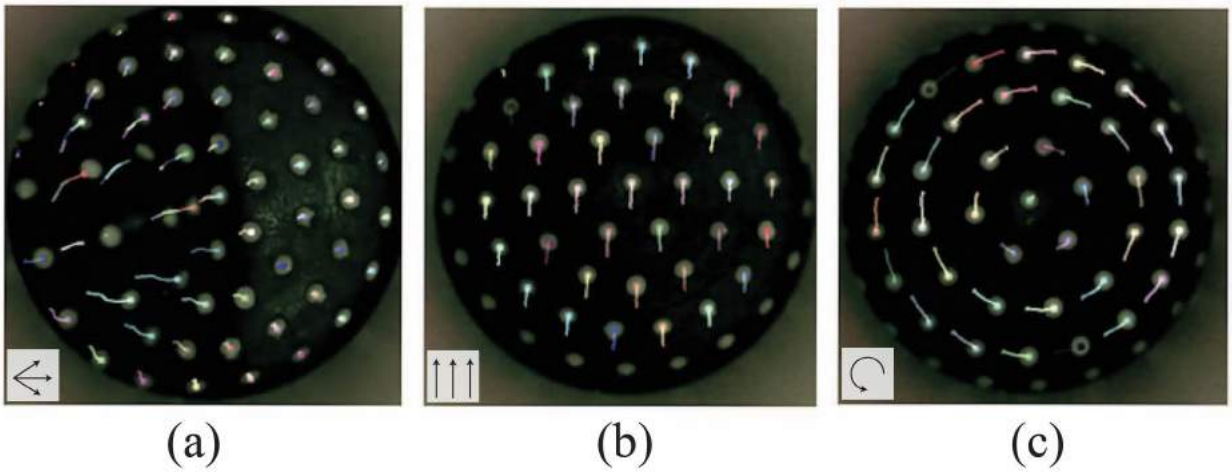


Figure 5.2: Examples of raw images captured by the internal camera, overlaid with the traces of the trajectories of the markers. Different colors are assigned to distinguish different marker traces. Arrows found in the bottom-left corners serve to illustrate the general directional motion of the markers. (a) The membrane is indented radially on the left side by a spherical probe (see Figure 5.4). (b) The membrane shears in contact with a flat surface moving upwards. (c) The membrane experiences torsion while in contact with a twisting flat surface. © 2018 IEEE

Tactile Image Processing

We rely on the internal camera data alone for tactile feedback. At first, we considered using pressure data from the MS5803-14BA sensor in parallel with the image data, as pressure has been used in some prior work to determine collisions of an inflatable robot arm [191]. However, through our own investigation, we determined that pressure signals show no substantial change that could robustly distinguish fine membrane perturbations such as those depicted in Figure 5.2, and thus are insufficient to be used for tactile data. We incorporate the pressure sensor in the device solely for device characterization.

We explored using common image processing techniques to extract relevant tactile information from the camera images. Our general preprocessing scheme began by low-pass filtering the image with a Gaussian blur, grey-scaling the image, and applying an Otsu adaptive threshold to binarize the image. Erosion and dilation was applied to remove false positive identifications of the cylindrical markers due to specular highlights, and further cleaning was performed using connected component analysis. OpenCV's `SimpleBlobDetector` was used to find the centroids of the white markers. The centroids were then tracked with OpenCV's implementation of optical flow for a sparse feature set using the Lucas-Kanade method with pyramids (`calcOpticalFlowPyrLK`).

From the optical flow output, we can calculate a set of vectors describing the translations of the markers from a previous frame. These vectors can then be used for dynamic tracking of the elastic membrane, inferring contact conditions, and detecting membrane shear. In Section 5.5,

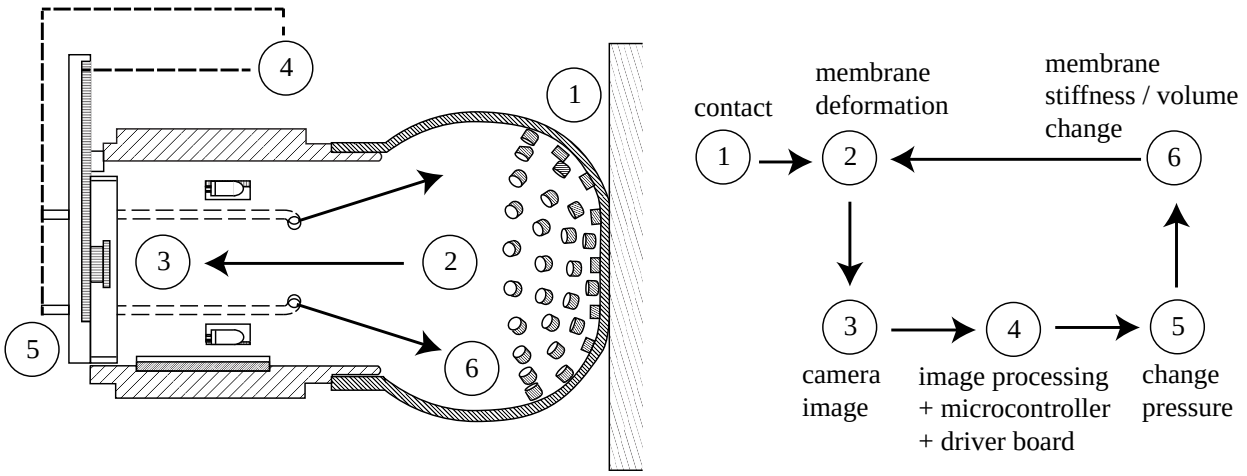


Figure 5.3: General system workflow. Contacts are transmitted as images via the internal camera and further modified by the image processing scheme. The vectors of marker displacements are then interpreted as tactile feedback, e.g., a detected membrane shear. A central computer computes a command, e.g., increase pressure, in response to the vectors, which is relayed to the microcontroller. The microcontroller sends a PWM signal to the solenoid valves, which in turn change the air pressure supplied to the SOFTcell. This subsequently changes the volume and stiffness of the elastic tip. The change in volume is captured by the internal camera as well, demonstrating proprioceptive capabilities of the SOFTcell. © 2018 IEEE

we demonstrate these sensing modalities at 30Hz in experiment. Figure 5.2 shows example trace vectors as the membrane is perturbed from the outside, and Figure 5.3 illustrates the workflow of the composite system.

5.3 Model of SOFTcell

We preface our experiments with a brief discussion of theoretical motivation. Intuitively, one expects that the SOFTcell membrane should become more rigid as internal pressure increases. The mechanical properties of SOFTcell are a function of both the thermodynamic state of the internal air and the strain state of the elastic membrane. Furthermore, contact forces and displacements of the membrane will depend on the contact geometry.

For small deformations about an equilibrium state, we assume that the mechanical response of SOFTcell can be modeled as that of an equivalent linearly elastic solid sphere subject to the same contact conditions. This equivalent sphere has an associated elastic modulus that we refer to as the effective modulus of SOFTcell, E^*. This effective modulus provides a useful predictive measure of SOFTcell's variable stiffness. The mechanics of hyperelastic materials under large deformations are complex and nonlinear. Previous work by Vella et. al. studied models of the indentation of

pressurized spherical shells, demonstrating the nonlinear relationship between effective modulus and internal pressure [203]. In 5.4, we show experimentally that the effective modulus provides a good approximation to the mechanical response of SOFTcell.

In our characterization experiments, we consider indentation of SOFTcell's membrane by simple geometries. From Hertz contact theory for the frictionless contact between elastic spheres, the load force F and indentation depth δ are related by [79]:

$$F = \frac{4}{3}\left(\frac{1-\nu_1}{E_1}+\frac{1-\nu_2}{E_2}\right)^{-1}\left(\frac{1}{R_1}+\frac{1}{R_2}\right)^{-1/2}\delta^{3/2} \tag{5.1}$$

where ν_i, E_i, and R_i are the Poisson's ratio, elastic modulus and radius for sphere $i \in \{1,2\}$. We will restrict our experiments to the case $E_1 << E_2$, that is, the deformation of the object in contact with SOFTcell is negligible. We can then define the effective modulus of SOFTcell under a specified pressure state and contact condition as,

$$E^* = \frac{3F}{4\delta^{3/2}}\left(\frac{1}{R_1}+\frac{1}{R_2}\right)^{\frac{1}{2}}(1-\nu) \tag{5.2}$$

Here, ν is the Poisson's ratio of the equivalent elastic sphere ($\nu = 0.5$ for ideal rubber). This result holds for plane contact in the limit as $R_2 \to \infty$.

5.4 Experimental Characterization

We characterize the variable stiffness and sensing capabilities of SOFTcell through a series of controlled experiments. In Section IV.A, we demonstrate the dependency of sensor mechanical stiffness, quantified here as effective modulus (Eqn. III.2) on internal pressure. In Section IV.B, we explore the force applied to a fixed object in contact with SOFTcell as internal pressure is periodically increased and decreased. Section IV.C summarizes our sensitivity analysis of SOFTcell's force sensing capabilities. All recorded values for pressures are gauge pressure.

Variable Stiffness

We use an ATI 45 six-axis force-torque transducer with a mounted 1.9 cm radius spherical probe to measure contact forces and torques on the SOFTcell. In these experiments, the SOFTcell is inflated to a fixed initial pressure and then quasistatically indented by the probe at a series of fixed indentation depths, both coaxially with the device and off axis by a 45 deg angle (Fig. 5.4) . Pressure and temperature within the cell are recorded, and the reaction forces and torques on the ATI sensor are recorded at mechanical and thermal equilibrium of SOFTcell. The results of these experiments are shown in Figure 5.5. We find that while the load forces on SOFTcell show similar monotonic increases with indentation depth, higher pressures result in larger load forces for the same indentation. The coaxial contact experiments are repeated with a planar probe.

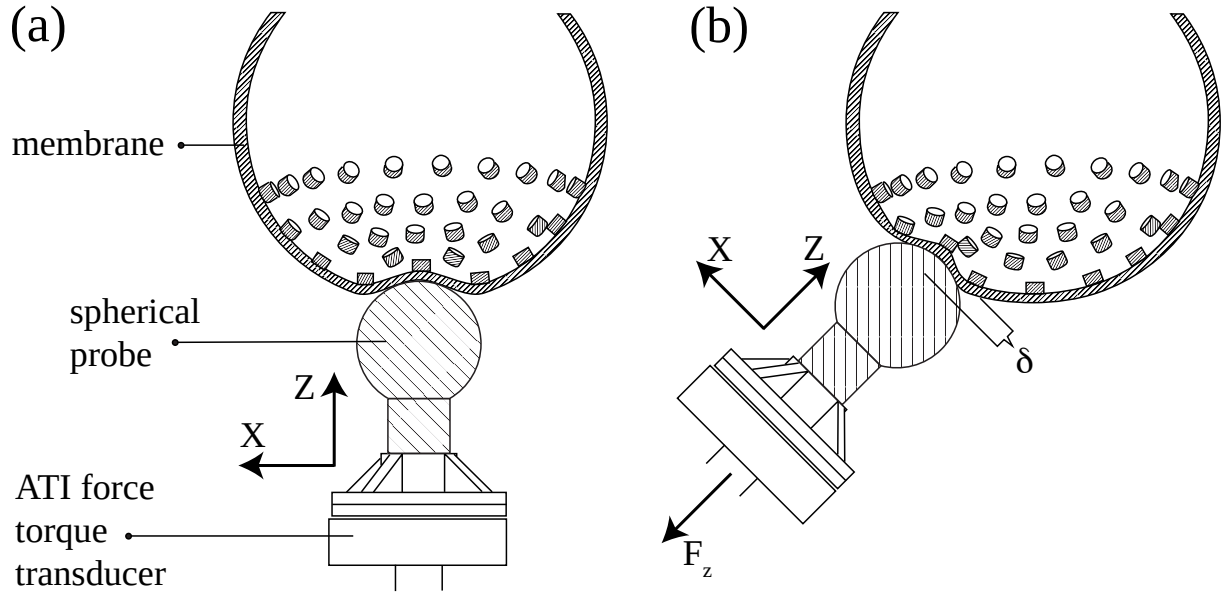


Figure 5.4: A schematic representation of the mechanical stiffness characterization experiments. (a) Coaxial contact with the spherical probe. (b) Radial contact at an angle of $\frac{\pi}{4}$. © 2018 IEEE

Following Eqn. III.2, we compute the effective moduli for SOFTcell at different internal pressure states. The results are shown in Figure 5.5. As hypothesized, E^* increases with pressure for all contact scenarios.

Dynamic Loading and Hysteresis

Using the same experimental apparatus as in the previous section, we measure the time-varying force imparted on a fixed probe as SOFTcell undergoes cyclic pressurization. The probe is placed so that it is initially in point contact with the membrane of SOFTcell so that there is no deformation of the membrane when the internal pressure of SOFTcell is the same as the ambient atmospheric pressure. The duty cycle of the solenoidal valve connected to SOFTcell is then increased and decreased from 0% to 100% as a triangle wave with a period of 17.5 sec (Fig. 5.6). The force measured on the load cell closely tracks the triangle wave signal, showing the responsiveness of SOFTcell's actuation. Furthermore, we find that after 222 cycles at 1 Hz, the load force remains consistent with a maximum peak of $1.17 \pm 1.66 \times 10^{-4}$ N, showing negligible hysteresis and demonstrating the repeatability of SOFTcell's actuation capabilities.

Sensitivity

We characterize the effect of controlling the internal pressure of SOFTcell on its sensing capabilities. For the previously decribed spherical probe, the minimum detectable coaxial deflection

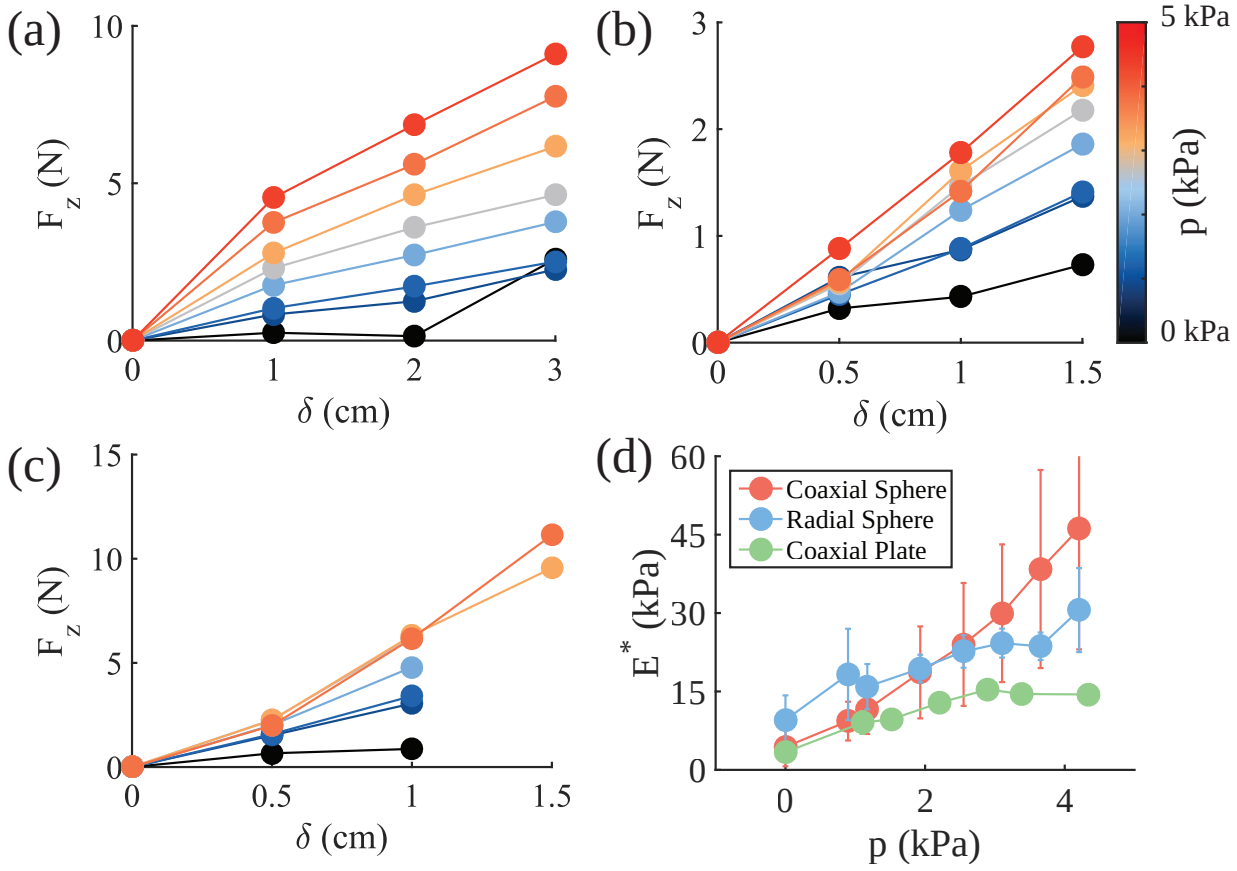


Figure 5.5: The results of the mechanical stiffness characterization experiments described in IV.A. (a) Coaxial contact with a spherical probe. (b) Radial contact ($\frac{\pi}{4}$) with a spherical probe. (c) Coaxial contact with a planar probe. At low pressures and higher indentations δ, the planar probe contacted the rim of the pressure capsule, so forces were not recorded. (d) SOFTcell's effective modulus increases as the internal pressure is increased. © 2018 IEEE

of SOFTcell above the camera's static noise is found to be 0.6 mm. For a small deflection, we linearize our empirical measurement of $F_z(\delta)$ (Fig. 5.5) at the origin to estimate minimum force resolution. We find that minimum detectable force increases with internal pressure so that smaller contact forces are detectable at lower pressures (Fig. 5.7).

5.5 Towards Reactive Grasping

We now utilize a different experimental setup as illustrated in Figure 5.8 in order to achieve two goals. We first aim to validate the Hertz contact and effective modulus model of SOFTcell through a simple shear experiment in Section 5.5, and we follow this with an implementation towards

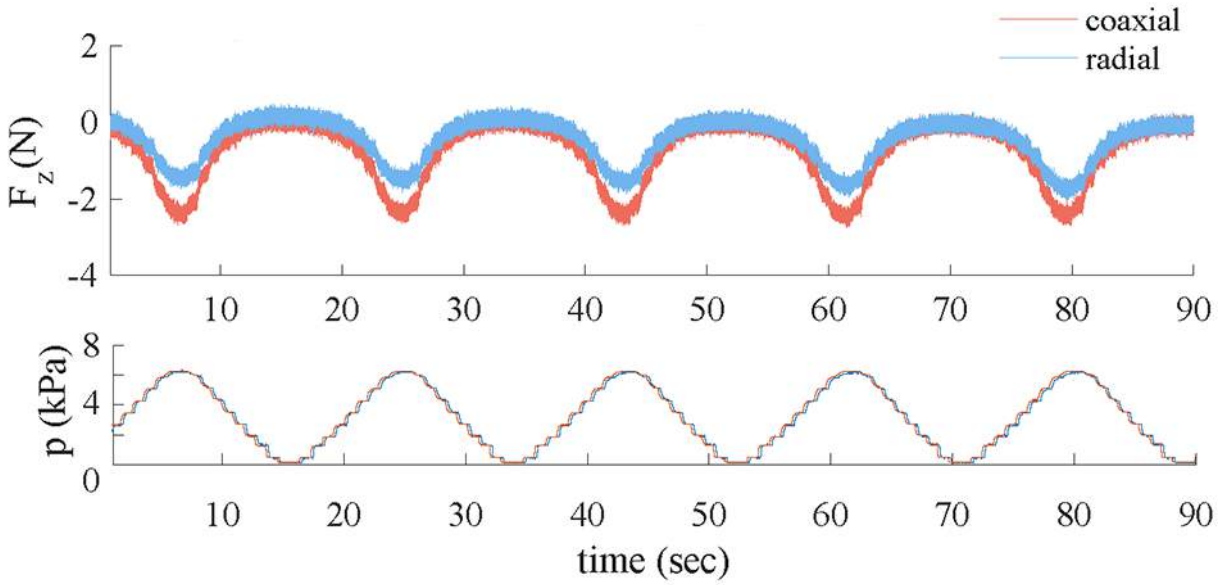


Figure 5.6: Comparison of coaxial and radial contact forces exerted by SOFTcell onto a fixed force-torque sensor. The internal gauge pressure is varied between 0 and 6.14 kPa in a triangle wave-like oscillation. © 2018 IEEE

simple reactive grasping in Section 5.5.

Model Validation

For this experimental setup, we fix SOFTcell such that its tip, under atmospheric pressure, is 20.25 mm from a vertical acrylic wall (see Figure 5.8). An empty 28.37 mm wide by 76.2 mm $\times$ 76.2 mm acrylic box with mass of 31.6 g is suspended between the uninflated SOFTcell and the wall, causing SOFTcell's initial normal membrane deflection (δ) to be 8.12 mm. SOFTcell maintains planar contact with the box throughout the experiment. The forces from the elastic membrane are sufficient at internal atmospheric pressure to counter the weight of the empty box.

We vary the internal pressure state of SOFTcell $(0-3$ kPa). For each pressure state, we continuously increase the mass of the empty box by adding 0.8 mm diameter / 670.2 μg glass beads. As the load increases, the gravitational force overcomes the static frictional force between the box and the wall, causing it to slip and the SOFTcell membrane to shear. We distinguish membrane shear from other perturbations such as pure indentation through the displacement vectors of the internal markers. Our heuristic determines shear if the standard deviation of the angular displacement is below an experimentally-determined conservative threshold (i.e., the displacement vectors are all within some margin in the same direction). Once a shear in the membrane is detected, we measure the mass of the box and repeat this process for a total of ten trials per pressure state (Fig. 5.11).

To visualize membrane shear displacement, we use SciPy's radial basis function interpolation class to approximate a denser map of displacements from the sparse set of displacements extracted

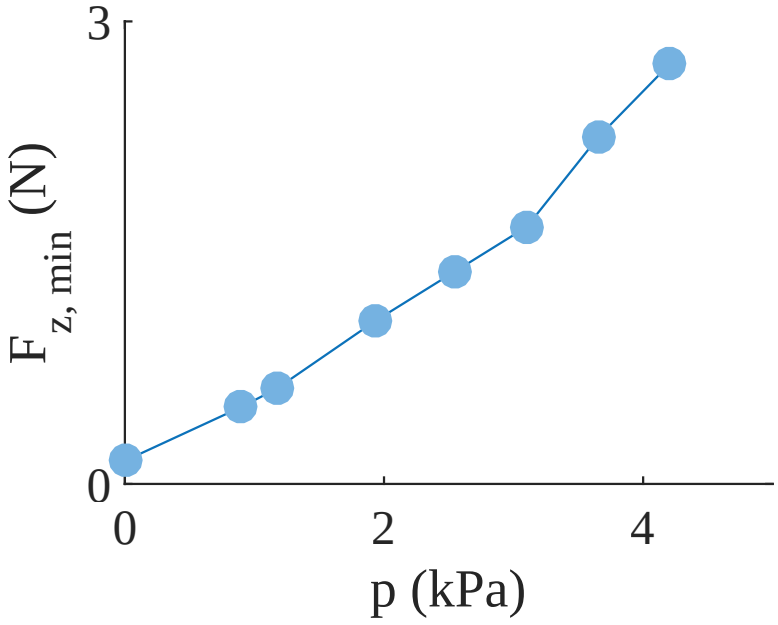


Figure 5.7: The minimum detectable force ($F_{z,min}$) for coaxial contact with the spherical probe vs. the internal pressure of SOFTcell. © 2018 IEEE

from the image. A contour plot of isodisplacement level curves can then be calculated from this dense map of interpolated displacements, providing an approximation of the global membrane motion and deformation. Figure 5.9 depicts representative isodisplacement contour plots of the membrane shear data measured by SOFTcell during the experiments. Each plot depicts sensor data of membrane deflection in the direction of shear under a shear load of $0.8N$. The displacement fields of the membrane in the direction normal to shear were found to be negligible. This is consistent with our assumption motivating the heuristic for shear detection that the direction of shear should be relatively uniform between the markers for a planar coaxial contact.

In addition to visualizing membrane shear from the displacement data, we also plotted the spatial average shear deflections over the membrane against shear load at varying internal pressures in Figure 5.10. As the shear load increases, SOFTcell detects larger shear deflections. This illustrates, from the clustered data of Figure 5.10 at different pressure states, given a measured average shear deflection, approximate shear force or load can be inferred.

Finally, from our measurements of box mass at the instance of slip, we estimate the normal force between SOFTcell and the box (F_N). We measure the normal deflection (δ) of the membrane at the center of contact and plot these results in Figure 5.11. From the Hertz contact model (Eq. III.1), we obtain a prediction for the normal force-deflection relation:

$$F_N(p,\delta) = \frac{4}{6} R(p)^{\frac{1}{2}} E^*(p) \delta^{3/2} \tag{5.3}$$

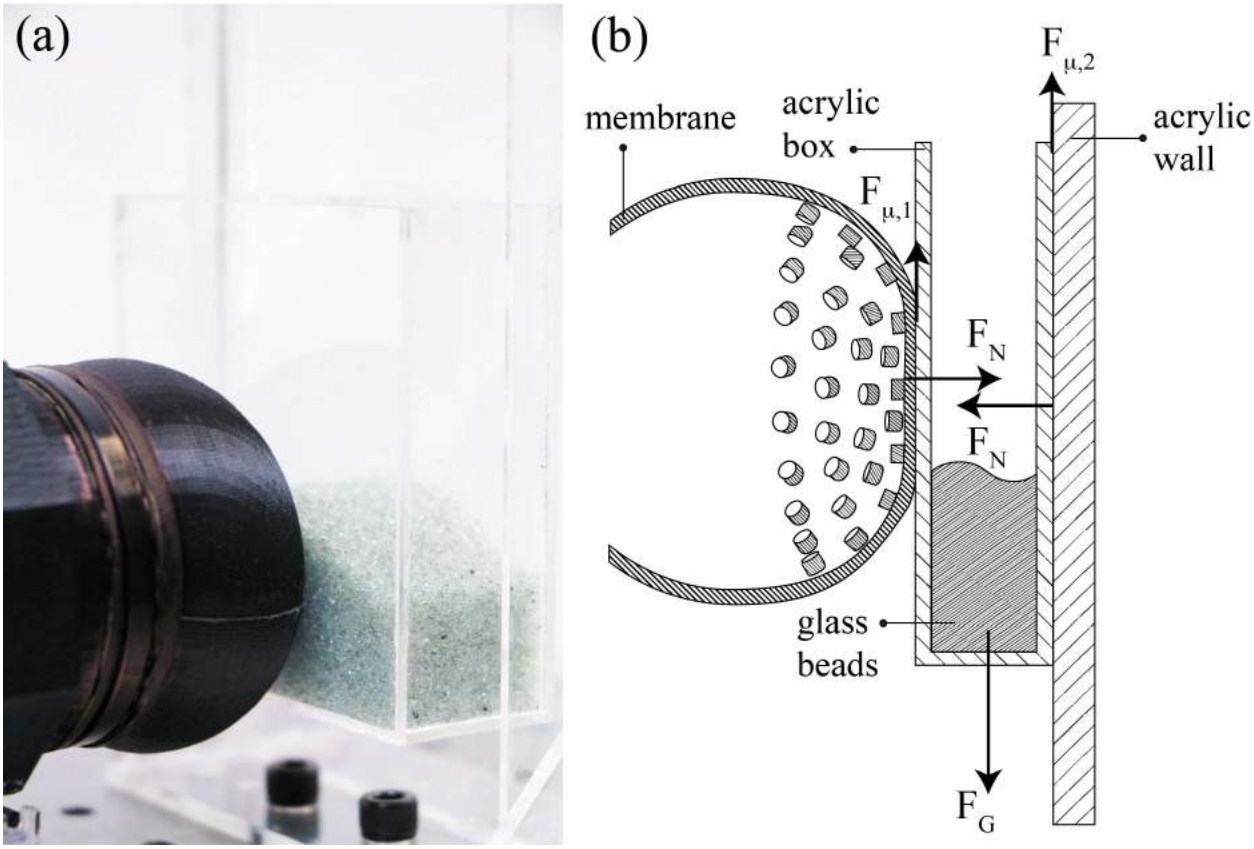


Figure 5.8: The experimental setup for Section 5.5 and Section 5.5. (a) SOFTcell pushes a box of glass beads against a wall, applying sufficient normal force to counter the effect of gravity; (b) a free-body diagram of forces involved in this experimental setup. © 2018 IEEE

where $R(p)$ is the tip radius at pressure p, and $E^*(p)$ is the effective modulus, determined empirically through characterization experiments (Fig. 5.5). We find close agreement between our Hertz contact and effective modulus model and empirical results. Deviations between the model and experiment are likely due to nonlinear effects outside of the regime of Hertz contact theory.

Task Demonstration

We present preliminary investigation of the potential for SOFTcell to aid in fine manipulation tasks, in particular, with reactive grasping. Specifically, we utilize SOFTcell's spatiotemporal sensing modality to extract contact conditions during a grasp. Membrane shear is measured by the sensor, and the actuation modality is then used to perform a reactive single-finger "grasping" adjustment of an object under varying load.

Measuring shear and slip of objects has been well-studied throughout tactile sensing literature

([223, 68, 137, 67, 187, 173, 193, 84]) and a few groups have incorporated this detection into closed-loop control for grasping ([187, 137, 173, 193, 84]). Here, we take inspiration from the experiments in [68] and [223] to simulate a grasp between a finger equipped with sensing (SOFTcell) and a stationary wall or "finger" that provides structural support, using the identical experimental setup in Section 5.5 (Figure 5.8). SOFTcell's actuation capability allows it to additionally adjust the strength of the grasp based on inferred grasp quality from sensory data.

For this demonstration, we begin with the same initial conditions as in Section 5.5, with an internal pressure state matching atmospheric pressure. SOFTcell is at an identical distance from the acrylic wall as before, and the same acrylic box is suspended between SOFTcell and the wall. We vary the load by pouring glass beads into the box, similar to an experiment presented in [84]. It is expected that at some critical mass (dependent on the internal pressure of SOFTcell), the membrane will shear as the box transitions from static to kinetic friction contact with the acrylic wall. When a shear is detected, the pressure within SOFTcell is increased to re-grasp the object and prevent slip.

As we vary the load of the box, SOFTcell successfully fine-tunes the grasp in real-time by incrementally increasing its internal pressure and subsequently the applied normal force to the box upon each detection of a shear, countering the new gravitational loads. We set the maximum internal pressure at $4kPa$ so that SOFTcell's membrane would not contact anything other than the box (as it grows larger, it has the potential to touch the platform on which it is mounted). At the maximum internal pressure of $4kPa$, SOFTcell is able to "grasp" a full box of glass beads with a mass of 210.2 g. A demonstration of this reactive grasping task is captured in the corresponding video to this chapter: `https://youtu.be/hipwkohfuYI`

5.6 Conclusions

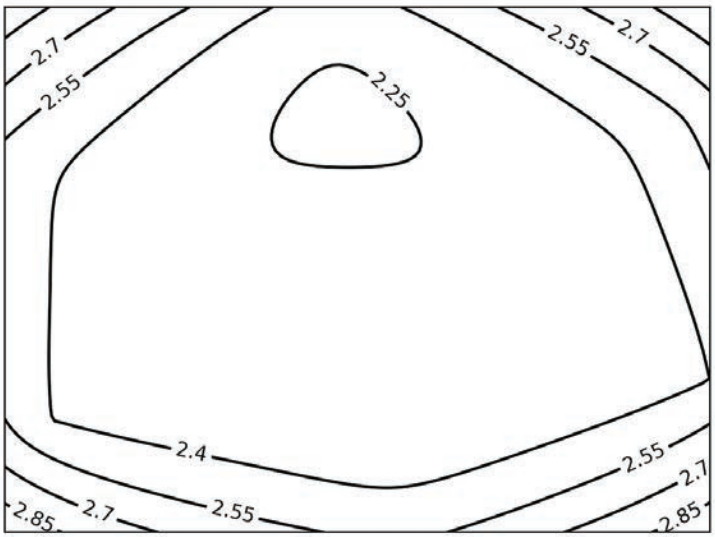
In this chapter, we presented the design and fabrication of SOFTcell, a low-cost variable stiffness pneumatic device capable of multimodal tactile sensing and actuation. We demonstrated that through a single control input (the pressure state within SOFTcell), we can control both the range of contact forces that can be detected through tactile sensing and the effective modulus for actuation. We also demonstrated how control of internal pressure allows for SOFTcell to operate multifunctionally through different sensing and actuation primitives, such as through increased sensitivity at lower pressure states and increased grasping force at higher pressure states. We developed a simple effective modulus model of the device based on Hertz contact theory and validated the model empirically. This effective modulus model enables estimation of the relationship between contact force and membrane strain depending on the cell pressure and contact conditions. Finally, we investigate the utility of SOFTcell in a reactive grasping task, demonstrating shear detection and variable stiffness to improve contact.

There are numerous possible directions for future work. One direct application of the SOFTcell would be in active tactile object and environmental exploration. Figures 5.2 and 5.9 provide examples of the rich image data that can be acquired from observing interactions of the elastic membrane with the environment. Since the SOFTcell was designed to be scalable, it could be miniaturized

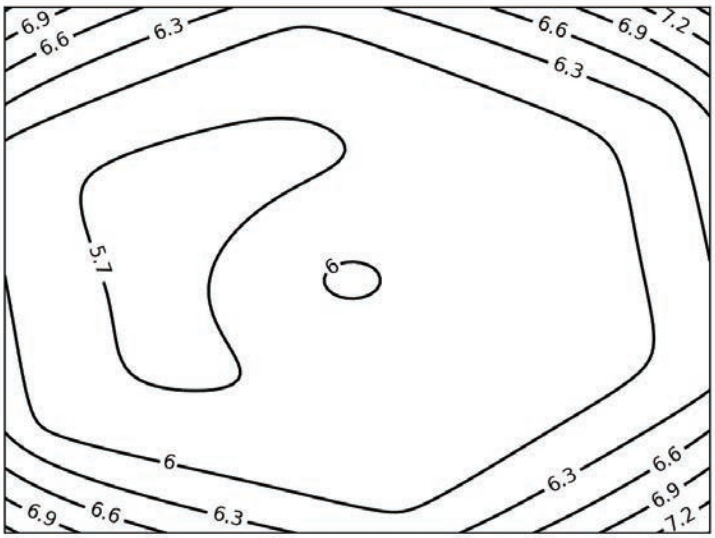
into a fingertip, although its size would ultimately be constrained by the parameters of the internal camera. With this data, one could explore basic tactile sensing tasks such as detecting incipient slip [68], categorizing material textures, tracing seams and object boundaries [56], reconstructing object geometries, and estimating material stiffness through impedance matching. It would be important to also characterize the resolution of details captured by the images when varying the stiffness of the membrane, as this would evaluate how SOFTcell augments its multimodal tactile capabilities through variable stiffness.

Furthermore, we imagine that the SOFTcell could be re-purposed as an end effector for legged locomotion. Substrate properties vary dramatically in natural environments, necessitating multifunctional appendages for legged locomotion. Using SOFTcell as a robotic appendage could improve robot mobility by controlling stiffness to actively explore surface materials [186] and to help gain traction in challenging terrain.

This chapter explored the sensing-actuation duality inherent in a device like the SOFTcell, demonstrating its capabilities in a manipulation task. However, we acknowledge that, because this device is pneumatically actuated, it cannot exist independently without being tethered to a larger, pneumatic system. Thus, we designed a different variable-stiffness tactile device in the following chapter, one that does not require tethering and is therefore easier to integrate with a robotic arm.



p = 0.34 kPa



p = 2.21 kPa

Figure 5.9: Contours of uniform membrane displacement from SOFTcell sensor data for a 0.8*N* shear load at different cell pressure states. Displacements are in *mm*, in the direction of shear. Displacements in the direction normal to shear were negligible for all trials.

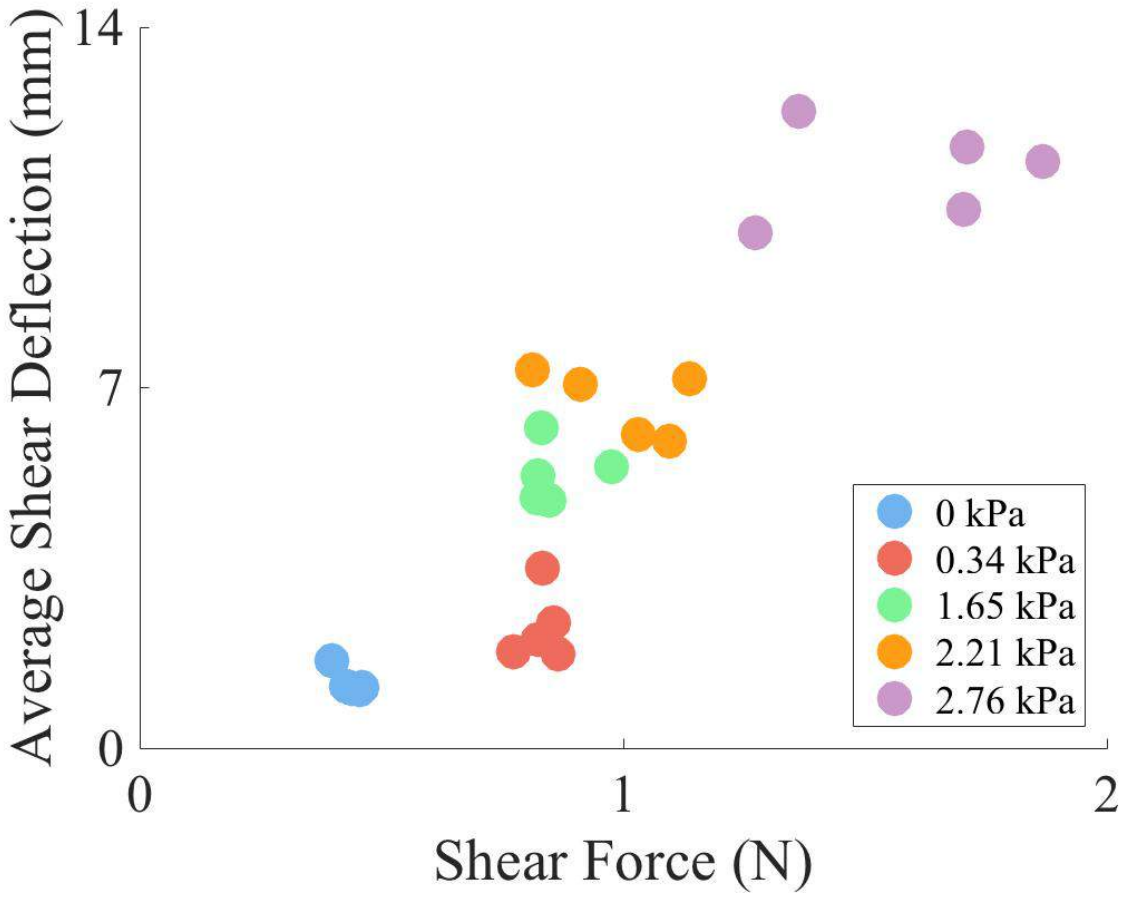


Figure 5.10: Average membrane shear deflection measured by SOFTcell for different shear loads at different cell pressure states. Different deflections are measured at different shear loads (depending on the internal pressure state), suggesting that approximate shear load can be inferred from the spatial tactile data. © 2018 IEEE

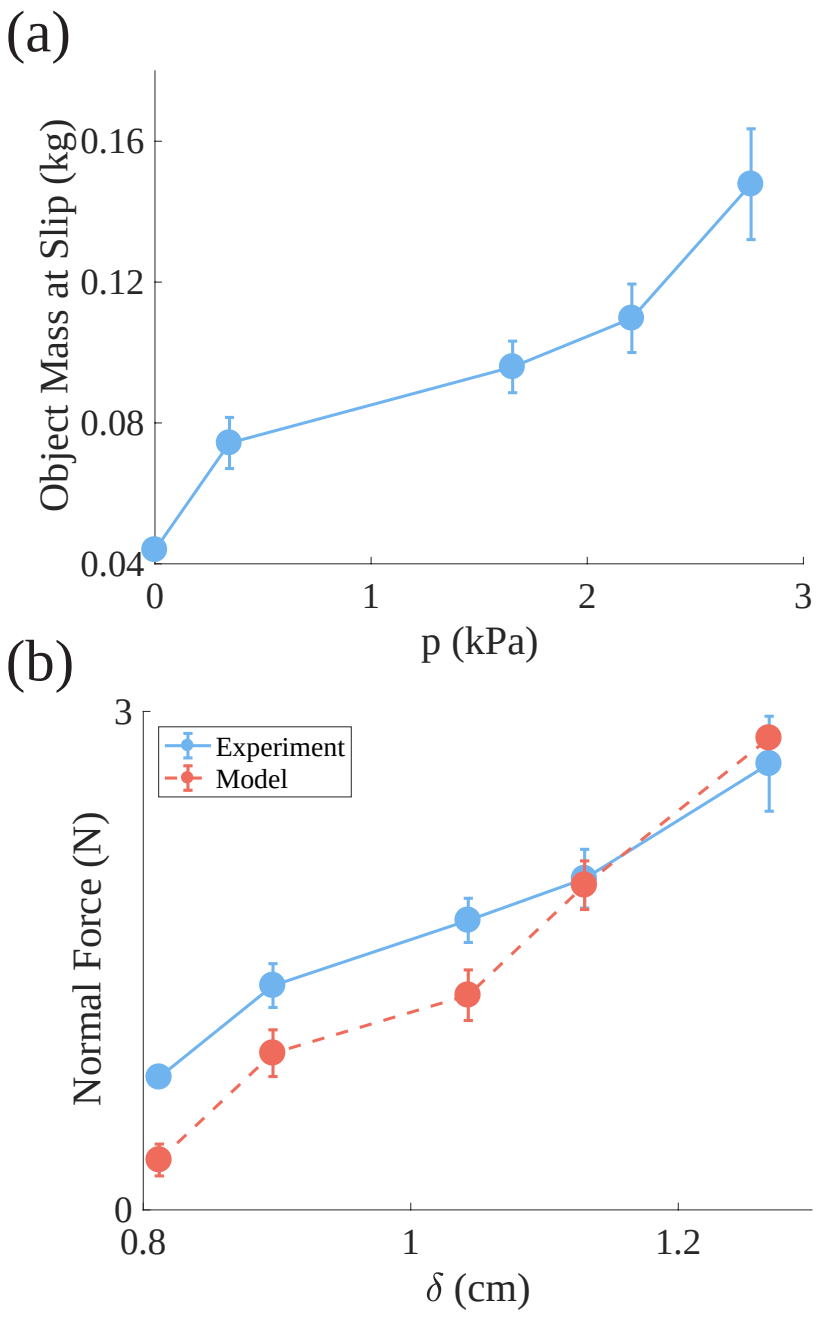


Figure 5.11: Results of the membrane shear experiments (Section IV.E). (a) The mass of the grasped object at the point of slip / detected shear vs. the pressure state of SOFTcell. (b) Measured normal force vs. normal deflection compared to theoretical predictions from the effective modulus Hertz contact model with measured parameters. Error bars show 95% confidence interval. © 2018 IEEE

Chapter 6

A Soft to Resistive Elastic Tactile Hand

This chapter is based on the paper "StRETcH: a Soft to Resistive Elastic Tactile Hand" [131], written in collaboration with Josephine Koe, and Ruzena Bajcsy.

Soft optical tactile sensors enable robots to manipulate deformable objects by capturing important features such as high-resolution contact geometry and estimations of object compliance. This work presents a variable stiffness soft tactile end-effector called StRETcH, a Soft to Resistive Elastic Tactile Hand, that is easily manufactured and integrated with a robotic arm. An elastic membrane is suspended between two robotic fingers, and a depth sensor capturing the deformations of the elastic membrane enables sub-millimeter accurate estimates of contact geometries. The parallel-jaw gripper varies the stiffness of the membrane by uniaxially stretching it, which controllably modulates StRETcH's effective modulus from approximately 4kPa to 9kPa. This work uses StRETcH to reconstruct the contact geometry of rigid and deformable objects, estimate the stiffness of four balloons filled with different substances, and manipulate dough into a desired shape. Experimental results can be viewed at `https://youtu.be/_kr5b5bs8y8`.

6.1 Introduction

With the advent of home robotics, robotic interaction with deformable objects such as electrical cables, bed sheets, and pizza dough is inevitable [177]. In particular, effective dexterous manipulation of solid deformable objects like dough requires the estimation of both contact geometry [220] as well as compliance [121]. This work aims to address the problem of estimating both stiffness and geometry while a robot is in contact with a solid deformable object.

To achieve this, this work proposes the development and use of a Soft to Resistive Elastic Tactile Hand, StRETcH. Different tactile sensors enable a robot to make various important measurements while in contact with an object, such as contact area, force, texture, and slip detection. Within the diverse set of existing tactile sensors, soft tactile sensors are particularly well-suited for interacting with deformable or fragile objects due to their inherent compliance. Prior work has demonstrated the clear benefits of soft material sensors, including sensitivity to stiffness for tumor

detection [206], high spatial resolution for classifying different textures [222], and handling of delicate objects like wine glasses [101]. The above mentioned soft tactile sensors are all optically-based, which enables high resolution contact imaging without the need to embed rigid components in the soft interface.

StRETcH is not only a soft optically-based tactile hand, but it is also able to adjust its stiffness and therefore vary the force it exerts on the environment. This work draws on our prior work in Chapter 5 where a soft dome-shaped fingertip called SOFTcell was developed, which explored the idea of a controllable-stiffness tactile sensor through pneumatic actuation [136]. An improved design of SOFTcell, which uses a depth sensor to image the deforming membrane [73], was later applied as an assistive robotic interface to dynamically support a human arm [71] as well as to perform geometry-dependent tasks [72]. In this work, we modify our prior sensor by designing the membrane to be planar rather than hemispherical, and varying its stiffness using a robotic gripper to mechanically stretch the membrane uniaxially instead of increasing its stiffness through pneumatic actuation. The benefits of this new design include the simple manufacturing process of the end-effector and easier control of the stiffness of the membrane via mechanical stretching rather than pneumatic actuation. This work aims to utilize the variable stiffness feature of the end-effector to estimate stiffnesses of different deformable objects as well as to modulate the force exerted on a solid deformable object in a manipulation task.

The main contributions of this chapter are:

- The Soft to Resistive Elastic Tactile Hand (StRETcH), a new variable stiffness tactile end-effector that is easy to manufacture, integrate, and use with a robotic arm.
- Modeling and experimental characterization of StRETcH and its sensitivity to contact area and object stiffness.
- A robotic demonstration using StRETcH to actively perceive and shape a piece of soft dough.

6.2 Related Work

Soft Tactile Sensing

Existing soft tactile sensors can be categorized by their underlying sensor technology – mainly, resistive, capacitive, piezoelectric, and optical sensing [220]. Many resistive-based sensors rely on intricate composition of individual nodes to achieve high-dimensional contact imaging [61, 110, 194, 44]. However, embedding rigid materials such as conductive wires within a soft skin limits the material's strain and therefore range of deformation. Thus, highly compliant fabric-like tactile materials [20, 6, 218] and ultra-stretchable capacitive sensors [105, 167] have been developed. These materials are largely useful for proprioception of soft robots [105] or soft exoskeletons [20, 6], but their low-dimensional signals are insufficient for relaying contact area information. While it has been shown that richer contact information can be extracted from much lower-dimensional

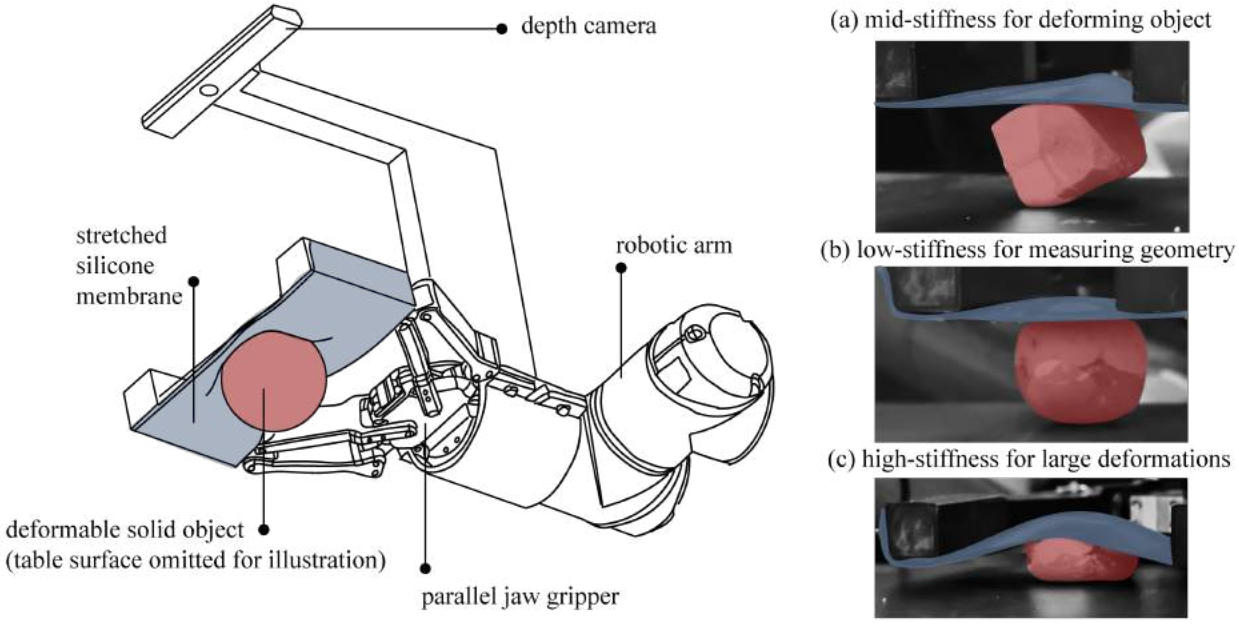


Figure 6.1: (Left): An orthographic view of StRETcH, mounted on a robotic arm. An elastic membrane (blue) is attached to the robotic gripper, which modulates its stiffness through stretching when interacting with an object (red). (Right): (a) Mid-range stiffness is used to deform a soft object; (b) Low-range stiffness is used when measuring the object's geometry; (c) High-range stiffness is used to exert large forces on the object. © 2021 IEEE

tactile signals [150], achieving this requires sophisticated modeling of the sensor as well as an extensive dataset relating raw signals to known geometric values.

Recent developments in optical image-based soft tactile sensors decouple the soft interface from the sensing mechanism, thereby achieving both high deformability and high contact resolution. Examples of such sensors include FingerVision [210], GelSight [224], TacTip [206], SOFTcell [136], and Soft-bubble [2]. In particular, the SOFTcell design from Chapter 5 incorporated both optical sensing and pneumatic actuation. Specifically, SOFTcell consisted of an elastic hemispherical membrane, which was imaged from within by an RGB camera and modulated in stiffness via pneumatic pressurization. This design enabled the sensor to be used as a controllable stiffness tactile device and explored the inherent sensing-action duality. Similar designs that use depth sensing to image the membrane were later developed in [73] and [2]. This work modifies the design of SOFTcell [136] for ease of manufacture, integration, and use with a robotic arm. We aim to use its controllable variable stiffness to enable high resolution contact imaging as well as measure deformable object stiffness.

Tactile Perception of Deformable Objects

One particular advantage we would like to exploit with the variable-stiffness image-based hand design is that it allows for high-resolution contact geometry and stiffness estimation of a wide range of deformable objects. Conventional tactile sensors often enable measurements for one or the other, but rarely both. This is because stiffness estimates are typically made by measuring the displacement of a probe upon application of a known force [184, 225], active palpation [44], or via piezoelectric resonance [82, 158], which inherently are low-dimensional signals and cannot replicate the rich geometric information available in image-based sensors. Meanwhile, soft optical tactile sensors, while able to image high-dimensional contacts, are typically only capable of sensing a small range of stiffnesses before deforming the object itself. For example, a soft optical tactile sensor called GelSight was used to estimate the hardness of several objects via active palpation [224]. However, the authors note that estimates of hardness are more accurate when the object hardness is close to the hardness of that of the sensor. Because the GelSight is a gel-filled tactile sensor, its elastic membrane has fixed stiffness and thus its sensitivity is limited to a narrower range of deformable objects. This work aims to address this limitation by enabling the elastic interface to vary in stiffness.

Manipulation of Deformable Objects

This work makes use of StRETcH's variable-stiffness to manipulate deformable objects into a desired shape. Deformable object manipulation can be categorized into the manipulation of deformable linear objects, planar or cloth-like objects, and solid objects [177]. Examples of linear object manipulation include rope manipulation [214] and knot tying [195], while planar object manipulation includes the folding of paper [8] or cloth [192, 139].

StRETcH was designed for the manipulation of solid deformable materials such as clay or foam. Prior work in solid deformable object manipulation depended primarily on image-based feedback. For example, in [153, 152], a bilateral manipulator uses features extracted from a stereo-camera and an estimation of deformation properties to shape unstructured soft objects like foam into desired two-dimensional shapes. The authors in [183] take a sensor-less approach by designing a unique morphing skin that wraps around sculptable materials like clay to deform the material. In [51], the authors pair a depth sensor with a force sensor attached to a rigid roller to flatten pizza dough into a circle. StRETcH incorporates a soft morphing skin as in [183] while maintaining a sense of contact force, as shown in [51]. When deforming solid materials, tactile sensing plays a large role in determining the shape and hardness of an object, making StRETcH appropriate for such applications.

6.3 System Design

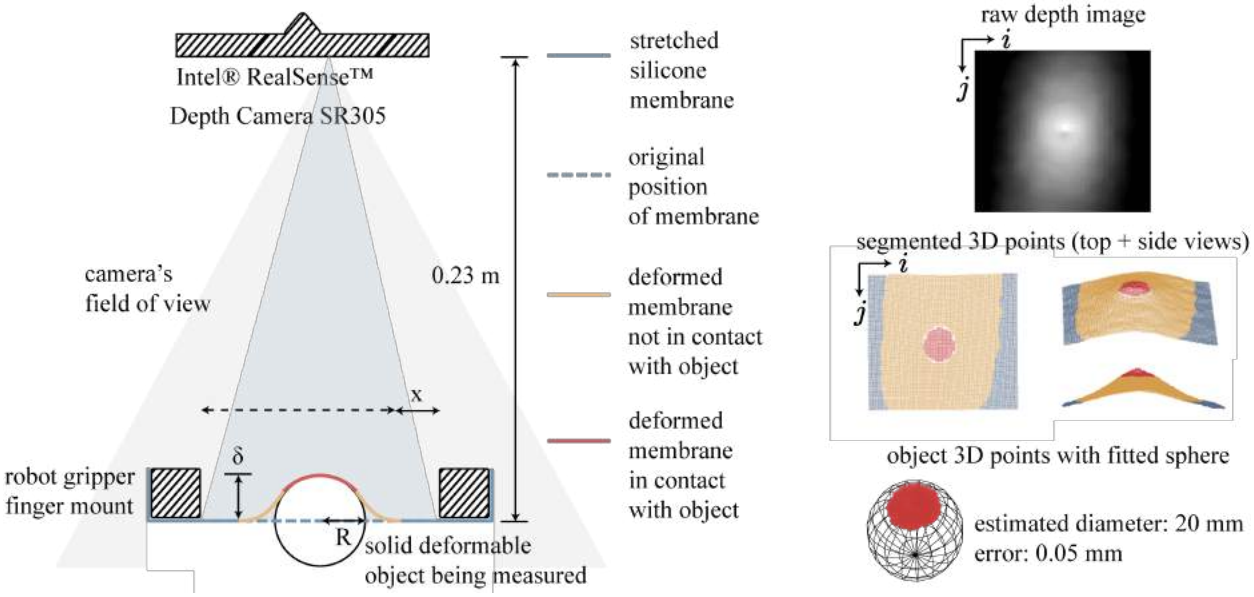


Figure 6.2: (Left): A cross-sectional schematic of StRETcH. The membrane is stretched by x mm. When the membrane makes contact with an object, it is deformed from its original position (dashed blue) and can be partitioned into undeformed (solid blue), deformed but not in contact (yellow), and deformed and in contact (red) segments. (Right): 3D points corresponding to the red, yellow, and blue segments are extracted from the depth image, resulting in a low error in estimated contact geometry. © 2021 IEEE

Design and Fabrication

StRETcH is comprised of an Intel RealSense Depth Camera SR305 and a flat silicone membrane attached to a Robotiq 2-Finger Adaptive Robot Gripper, which are both fixed to a Universal Robot UR5 robotic arm (see Figure 6.1). The parallel-jaw gripper serves as the stretching mechanism for the membrane as it varies its gripper throw and can stretch the membrane up to $x = 80$mm past its original length (see Figure 6.2). The silicone membrane was manufactured by pouring platinum-catalyzed EcoFlex 50 Silicone into a flat mold, which produced a 100mm $\times$ 90mm $\times$ 2mm rectangular membrane. Velcro was sewn onto strips of cloth that were embedded into the sides of the silicone during the curing stage. The silicone membrane was attached to 3D-printed gripper finger mounts with velcro.

Contact Estimation

To interpret the deformations of the membrane captured by the depth sensor, we must first understand that the membrane surface can be categorically divided by its deformation state. Specifically, we follow the classification defined in [73] to segment the membrane into the following categories based on the estimated curvature at each point:

(a) Undeformed: the membrane lies flat at its initial depth and is assumed to not be in contact with the object.

(b) Deformed but not in contact: the membrane is deformed but not convex along either axis of the 2D depth image.

(c) Deformed and in contact: the membrane is deformed and is convex in either axis of the depth image.

The segmentation of category (c) points enables estimation of contact area and geometry, as shown in red in Figure 6.2. Segmentation is based off of the proposed algorithm in [73], which assumes that the object in contact with the membrane is flat or convex, and the contact area of the object is larger than a single surface element of the membrane.

Under these assumptions, membrane segmentation begins by first isolating the membrane in the depth image. This is done by using a box mask that depends on the width of the gripper throw. The depth image points corresponding to the membrane are projected into 3D and transformed into coordinates relative to the membrane plane when it is flat and undeformed. Normal vectors are then estimated from the 3D point cloud of the membrane. These normal vectors and points are used to calculate curvature at each point. Let depth image pixel (i, j) have 3D coordinates $\vec{p}(i, j)$ and normal vector $\vec{n}(i, j)$. Local curvature signals $\mathcal{K}_i$ and $\mathcal{K}_j$ along the i and j directions, respectively, can then be estimated at each depth image pixel using the method presented in [151]:

$$\begin{aligned} \mathcal{K}_i(i,j) &= \langle \vec{p}(i+1,j) - \vec{p}(i-1,j), \vec{n}(i+1,j) - \vec{n}(i-1,j) \rangle \\ \mathcal{K}_j(i,j) &= \langle \vec{p}(i,j+1) - \vec{p}(i,j-1), \vec{n}(i,j+1) - \vec{n}(i,j-1) \rangle \end{aligned} \tag{6.1}$$

If a curvature signal is positive, the point is locally convex in that direction; if it is negative, it is locally concave. Thus, 3D points are categorized as *in contact* with the object if they exhibit sufficiently high curvature in either direction. Additionally, any points lying above the plane of these points are also considered to be in contact with the object, thus accounting for contacts with flat-faced objects.

6.4 Model of StRETcH

The StRETcH membrane becomes more rigid as it is stretched. Characterization experiments show that the mechanical response of StRETcH is a function of the internal strain state of the elastic membrane, caused by uniaxial stretching from the gripper and transverse loading and displacement

of the membrane from the object in contact, which depends on the contact geometry and depth of indentation.

The mechanics of Neo-Hookean membranes are complex and nonlinear, and, to the best of our knowledge, there does not exist an explicit model that calculates the mechanical response of a uniaxially stretched Neo-Hookean membrane under contact with elastic objects of varying geometries. There exist models for spherical [15, 216, 12] and cylindrical [159, 119] rigid indentation of stretched membranes, although these membranes are omnidirectionally stretched and our experimental data does not match their power laws. Recently, numerical simulations present an alternative to explicit models to, for example, calculate the strain due to uniaxial stress on rectangular sheets of rubber [198, 198]. However, numerical simulation is often too slow for real-time robotic use.

Instead, we borrow and revise from prior work [136, 182] the idea of modeling the membrane as an equivalent linearly elastic solid half-space subject to the same contact conditions. This model is not meant to represent the internal strains of the *membrane*, but rather, we adopt it as a template to express StRETcH's varying stiffness. We imagine that this equivalent half-space has an associated elastic modulus, which we refer to as the effective modulus E^*, to approximate the mechanical response of the membrane at its different stretch states.

Hertz contact theory [79] relates the load force F and indentation depth δ for frictionless contact between:

1. an elastic half-space and an elastic sphere:

$$F = \frac{4}{3}(\delta^3 R)^{1/2}\left(\frac{1-\nu^2}{E^*} + \frac{1-\nu_o^2}{E_o}\right)^{-1} \tag{6.2}$$

2. an elastic half-space and a rigid cylinder

$$F = 2R\delta E^* \tag{6.3}$$

3. an elastic half-space and a rigid cone

$$F = \frac{2\delta^2}{\pi(1-\nu^2)\tan(\theta))}E^* \tag{6.4}$$

where R is the radius of the sphere and cylinder, θ is the angle between the conical surface and the elastic surface, and E^* and ν are the effective elastic modulus and Poisson's ratio of the half-space. We use the Poisson's ratio of ideal rubber (0.5) for ν. E_o and ν_o are the elastic modulus and Poisson's ratio of the elastic sphere, but in characterization experiments, the indenters are rigid, so it is assumed that $E^* << E_o$. Characterization experiments measure the mechanical response of the membrane against all three basic geometries and in various stretch states and indentation depths. All contact loads are centered, transverse, and normal to the membrane surface. Despite mechanical differences between elastic membranes and half-spaces, this simplified representation enables fast and empirically accurate load force estimations (Section 6.5).

6.5 Experimental Characterization

Variable Stiffness

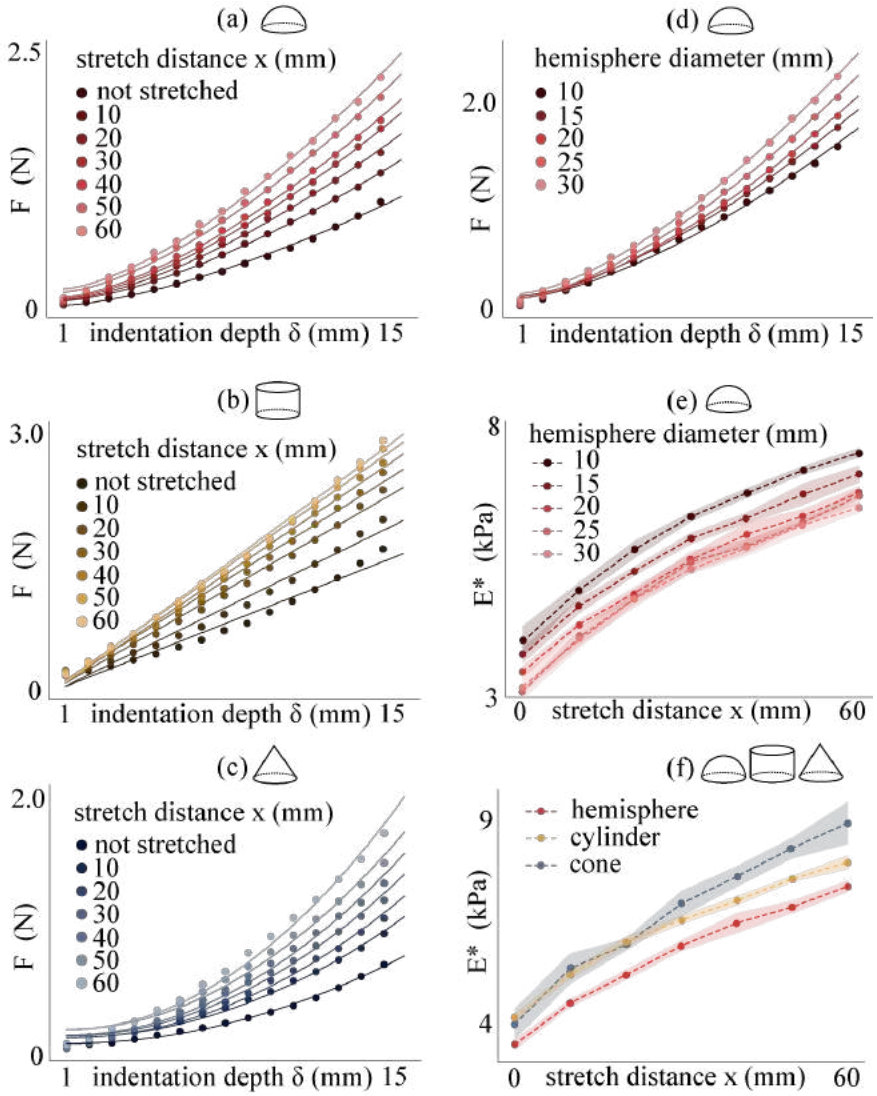


Figure 6.3: (a) Indentation depth δ vs. load force F for varying stretch states for a 20mm diameter hemispherical rigid indenter; (b) δ vs. F for varying stretch states for a 20mm diameter cylindrical rigid indenter; (c) δ vs. F for varying stretch states for a 45 deg conical rigid indenter; (d) δ vs. F for varying diameter hemispherical indenters with a membrane stretched by 30mm; (e) Stretch distance x vs. effective modulus E^* for varying diameter hemispherical indenters; (f) x vs. E^* for indenters of varying geometry.

To characterize the varying stiffness of StRETcH under different contact conditions, we measured the contact force of a rigid hemisphere, cylinder, and cone, as well as rigid hemispheres with different diameters, on the membrane at various uniaxial stretch states (x) and indentation depths (δ). A clamped ATI Axia80 EtherNet Force/Torque (F/T) sensor was used for force measurements, and 3D-printed probes of hemispheres, cylinders, and a cone were attached to the F/T sensor through friction coupling with a 3D-printed interface. The UR5 robot was commanded to press the membrane into the probe up to 15mm in depth in the z-axis of the F/T sensor, with 1mm steps, generating a vector of indentation depth to force data pairs. This was repeated 5 times each for 7 total stretch states, from zero applied tension to a Cauchy strain of 0.8 (the membrane was stretched up to 60mm beyond its original length). The dots in Figure 6.3 correspond to the force data measured at each contact condition.

Using equations from Section IV, an effective modulus (E^*) was fit for each stretch state to the pairs of force to indentation depth data. The calibrated model with the estimated effective modulus is shown as the solid lines in Figure 6.3. As shown in graphs (a)-(d), the resulting model adheres closely to the real data. Deviations occur for small indentations, but past 3mm indentations, the model performs well, especially under the condition of spherical and conical contact. Additionally, Figure 6.3(d) shows that as the contact area increases, the load force increases as well. Finally, the averaged estimated effective moduli E^* (in kPa) at each stretch state for the different scenarios (contact area and geometry) were plotted in Figure 6.3 (e) and (f), respectively. As shown in both graphs, the effective moduli monotonically increase in the operating region, which demonstrates that StRETcH indeed does have variable stiffness. The cylindrical and hemispherical data also show low variance over 5 trials, demonstrating that this variable stiffness is reasonably repeatable and controllable.

Contact Estimation Sensitivity

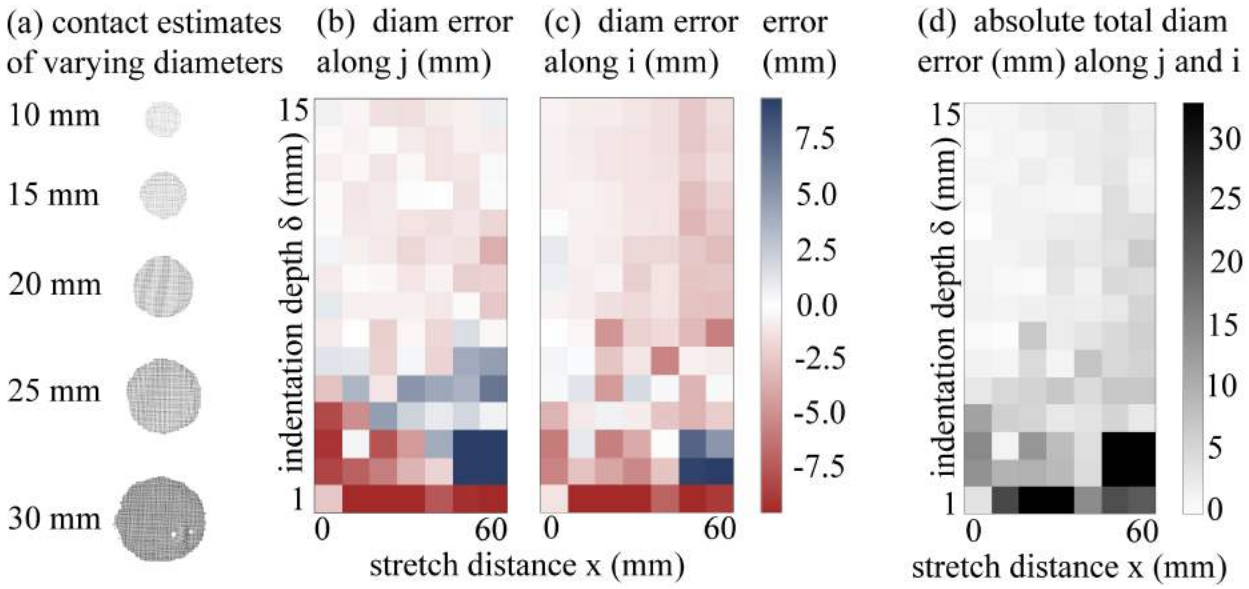


Figure 6.4: (a) Example estimated contacts with cylinders of varying diameters; (b) Errors in estimated diameter along the stretch direction j for different indentation depths δ and stretch distances x; (c) Errors in estimated diameter in the i direction for different δ and x; (d) Absolute total error in estimated diameter along the i& j directions for different δ and x. (b-d) demonstrate higher contact estimation accuracy for larger δ and lower x. © 2021 IEEE

In addition to characterizing StRETcH's variable stiffness, membrane indentation experiments were also performed to quantify the sensitivity of StRETcH in measuring contact geometry (see Section 6.3). Five right circular cylinder probes were used in this characterization, ranging from 10mm to 30mm in diameter. For each cylinder, the UR5 was again commanded to press the membrane into the probe up to 15mm, in the direction of the length of the cylinder, for the 7 stretch conditions characterized in the prior section. The error in estimated diameter along the i and j directions were calculated for each scenario and depicted in Figure 6.4.

As shown in Figure 6.4(a), the sensor is capable of sub-millimeter accuracy when estimating contact of the varying circular faces. However, this level of sensitivity arose under specific contact conditions. Figures 6.4(b) and (c) visualize the diameter error for the 20mm cylinder under different indentation depths and stretch distances in the i and j directions. As shown in these two plots, in general, the high-sensitivity region corresponds to larger indentation depths and lower stretch states. The high-accuracy region improved with decreasing diameter, since larger flat contact areas resulted in lower curvatures registered by the contact estimation algorithm. Indentation depths of less than 2mm tended to perform poorly for all five cylinders. Furthermore, it appears that the more the membrane is uniaxially stretched, the greater the indentation depth is needed to achieve

sub-millimeter accuracy. This matches the intuition that the stiffer the membrane is, the greater the force necessary for the membrane to conform to the geometry.

Estimating Load Force from Estimated Contact

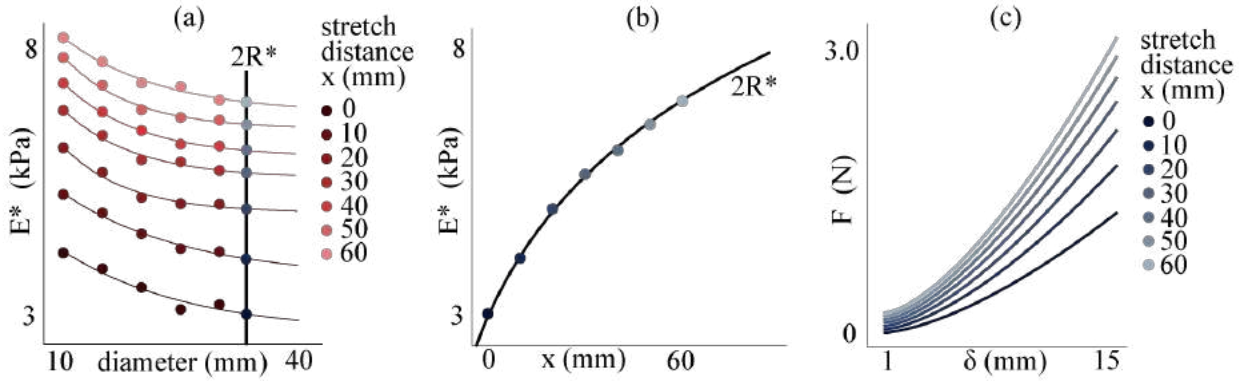


Figure 6.5: (a) Alternative visualization of Figure 6.3(e). Negative exponential curves are fitted to the datapoints (which are the average E^* over five trials). The vertical line denotes corresponding effective moduli for an estimated spherical contact with radius R^*. (b) A logarithmic function is fit to the corresponding points lying on the vertical line $2R^*$ in plot (a). The fitted function estimates E^* given the stretch distance x for the particular geometry of radius R^*. (c) Generated δ vs. F functions for the specific radius R^*. © 2021 IEEE

The above characterization experiments present StRETcH as a multimodal system: at low stretch states, the hand has high sensitivity to contact geometry, and higher stretch states enable greater contact forces. Here, we examine the accuracy of load force estimation given the estimated contact geometry at a particular indentation depth δ and stretch distance x.

Let us consider contact with a rigid spherical object that has an estimated radius of R^*. To estimate the contact force of StRETcH with the object, we need to derive a model of $E^*(x)$ for the specific radius R^*. First, note that the datapoints in Figure 6.3(e) can be equivalently represented as in Figure 6.5(a). For each stretch distance, a negative exponential curve is fit, where each curve is a function of the contact diameter. This enables estimations of E^* for a particular contact radius R^* at different stretch distances. A logarithmic function can then be fit (see Figure 6.5(b)) to model E^* for different stretch distances x, given the estimated radius R^*. Load force F^* is thus estimated by plugging E^*, R^* and δ into Equation 6.2. Figure 6.5(c) shows $\delta - F$ curves for R^* and various values of x, which are generated using the model in Figure 6.5(b).

We tested the accuracy of estimating F^* given the estimated radius R^* with leave-one-out cross-validation. Specifically, using four of the five curves in Figure 6.3(e) to fit the model for E^* given x and R^*, we tested the accuracy of the estimated F^* against ground truth for indentations of the left-out data. On average, the estimated load force had an error of $0.023 \pm 0.011N$, which is less than

1% of the total range of the operating region. Thus, using the model in Section 6.4, StRETcH's mechanical response can be estimated given contact conditions and stretch state.

6.6 Experiments

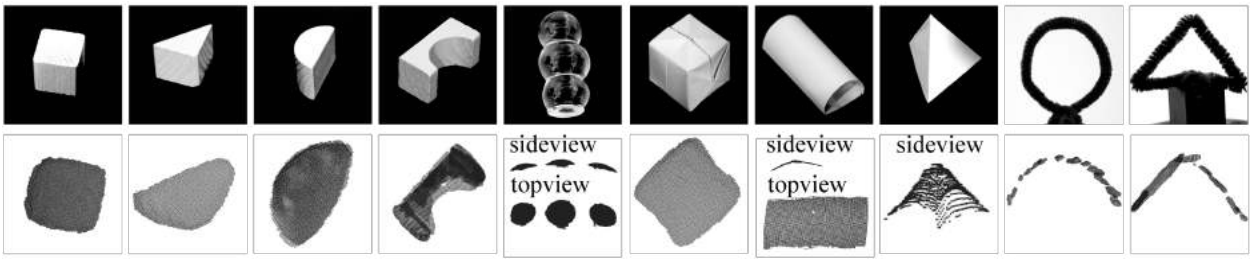


Figure 6.6: 3D reconstructions of contacts with a: wooden cube, wooden triangular prism, wooden half-cylinder, wooden bridge, glass snowman, paper cube, paper half-cylinder, paper pyramid, wire circle, and wire triangle. © 2021 IEEE

3D reconstruction of deformable and clear objects

To demonstrate the capabilities of StRETcH as a tactile hand, we first used it to reconstruct the geometry of several objects. As shown in Figure 6.6, StRETcH was capable of reconstructing the contact of several basic geometric shapes as well as soft enough to conform to highly deformable material such as paper and wire. When contact is made, the 3D points associated with the estimated contact area are transformed into the robot world coordinates. As the robot moves to collect a new observation through StRETcH, the object point cloud is updated with every new contact. This allows for the reconstruction of "multi-view" objects such as the wire circle. For this section, the robot was jogged to multiple positions if necessary (e.g, for the larger objects such as the wooden bridge, snowman, and wire objects). The membrane was kept at its lowest stretch state (zero applied tension) to maintain its high sensitivity and conformability in the case of encountering a deformable object.

The 3D reconstructed geometries match closely to the corresponding objects. From the wooden objects, it is clear that the membrane, at its low stretch state and indentation depth, causes the estimated geometry to be smoother, rounding out the corners and edges of the shapes. However, the dimensions of the contacts are within 2 mm of error from the real dimensions. The glass snowman, which would normally not be imaged properly through traditional vision, is found to have three domes. The paper 3D objects are imaged with accurate reconstruction, and, despite making contact with StRETcH, the heights of the paper objects are maintained, with an error of 3 mm. Finally, the robot collected multiple viewpoints of the highly deformable wire circle and triangle, reconstructing the original wire geometry.

Estimation of deformable object stiffness

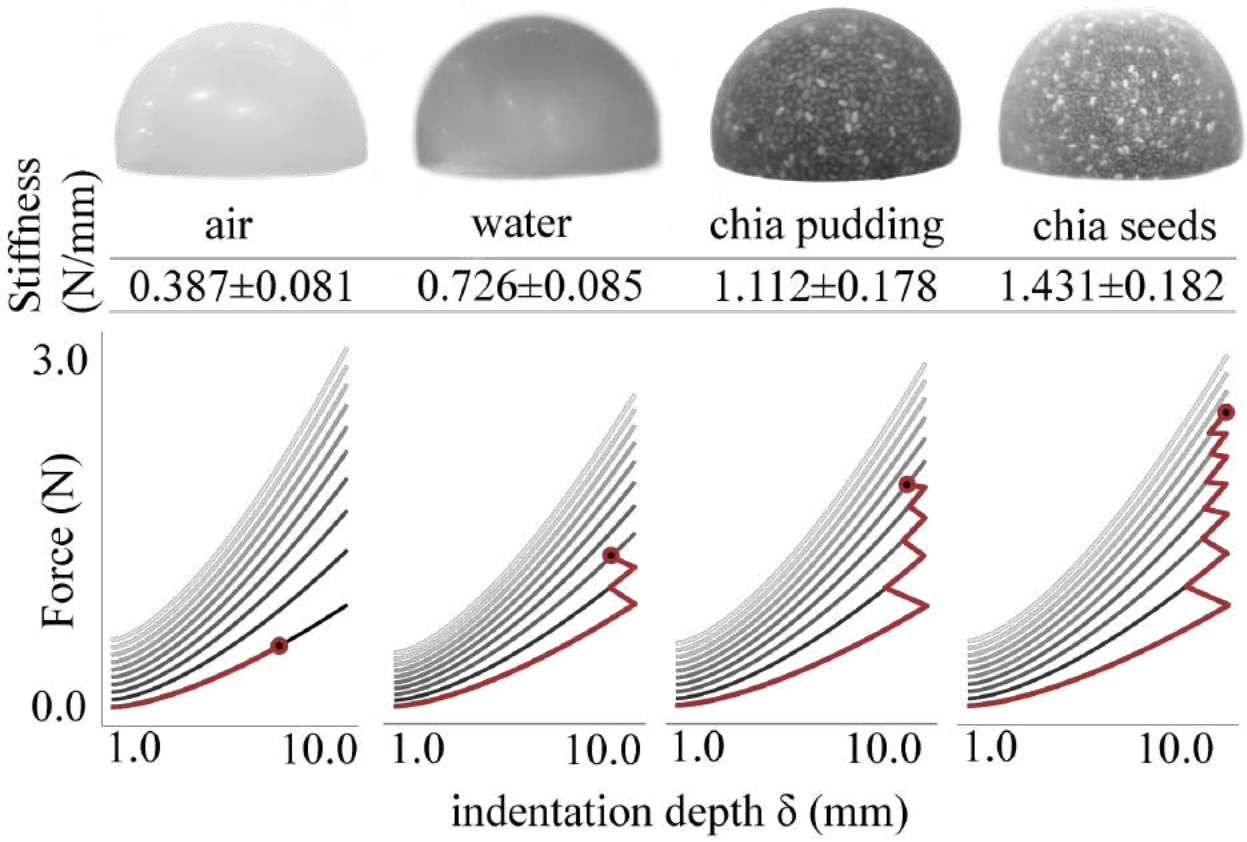


Figure 6.7: (Top): Stiffness estimations in N/mm for balloons filled with air, water, chia pudding, and chia seeds. (Bottom): Active sensing is used to determine stiffness. Plots display force modulation of StRETcH (shown in red) as it increases its indentation depth (moving along the curve to the right) and stretch width (moving to higher curves) until a deformation is sensed. δ-F curves are generated using the learned model described in Section 6.5. © 2021 IEEE

This section demonstrates how StRETcH actively interacts with a deformable object to estimate its stiffness. Four balloons were filled with approximately the same volume of air, water, a mixture of water and chia seeds, and chia seeds. When water and chia seeds combine, the mixture becomes very viscous relative to water. Qualitatively, the air balloon was the least stiff, followed by water, chia pudding, and chia seeds. Each balloon was fixed inside a cup so it would not move during palpation experiments. The exposed portions of the balloons were approximately hemispherically shaped.

To estimate stiffness, the robot algorithm begins by estimating the radius of the balloon. This is done using the lowest stiffness regime of StRETcH, which, as shown in Section 6.5, has greater sensitivity to geometry due to high conformability under zero uniaxial strain. The robot takes note

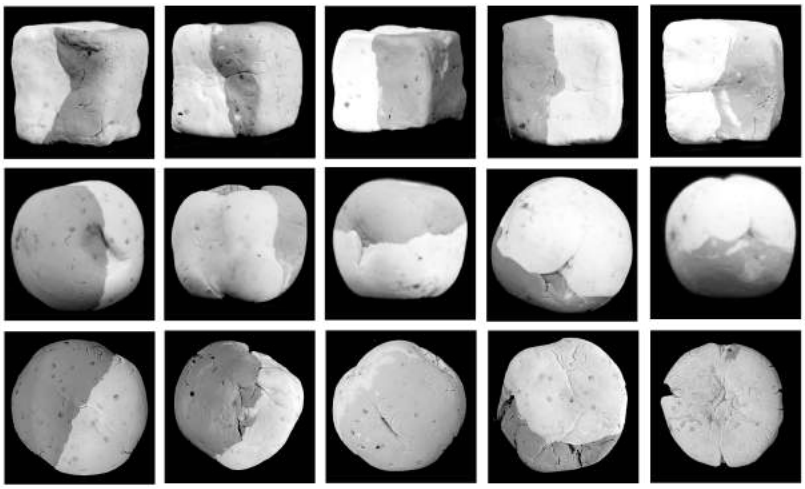
of the balloon's height and presses StRETcH's membrane into the balloon up to 10mm. After observing the contact at 10mm, a sphere is fit to the estimated contact points. Estimating the geometry is necessary, as Figure 6.3(e) shows that the effective modulus E^* varies with the contact area.

The robot proceeds to increase its stretch state incrementally and continues pressing the membrane into the balloon until the height of the balloon is sensed to have dropped below a certain threshold from its original height. We set this to 1.5mm as sensor noise was on the order of $\pm$1mm. At this point of deformation, the force F^* exerted by the membrane onto the deformable balloon can be calculated using the fitted model described in Section 6.5.

A proxy for stiffness in N/mm is then determined. Approximating the object as an elastic body with one degree of freedom, the stiffness is calculated by dividing F^* by the deflection of the balloon under the membrane's imposed force. As shown in the upper table of Figure 6.7, StRETcH is capable of repeatably estimating the stiffness of a wide range of substances. Note that, although these are technically proxy values for stiffness due to the simplified representation of the balloons, the estimated values are sufficient for differentiating between materials within this range.

Finally, Section 6.5's model can be used to improve the robot's active sensing capabilities. The lower plots of Figure 6.7 display example indentation depth (δ) to load force (F) curves at different stretch states (x), with higher curves corresponding to higher stretch states. The red lines denote the indentation depths and stretch states explored until a deformation is sensed for that particular object. As shown in the plots, if no deformation is sensed by StRETcH before the membrane is stretched any further, the model can be used to calculate the indentation depth to begin palpation for the next stretch state such that the force applied by the membrane is monotonically increasing until the deformation is sensed. This demonstrates that StRETcH can incorporate the model to actively adjust during system identification of the object in question.

Rolling a piece of dough and forming a sphere



circularity (pressed dough)	0.890	0.868	0.893	0.884	0.904

Figure 6.8: Example cookies formed from cubes of Play-Doh. (Top): The dough begins as a cube; (Middle): The dough is rolled into a sphere using medium stiffness; (Bottom): The dough is flattened into a cookie using high stiffness. © 2021 IEEE

We conclude with a demonstration of the use of StRETcH to form a cookie out of Play-Doh by rolling the dough until it forms a sphere and then pressing it into the shape of a cookie. This task necessitates three different stiffness regimes of StRETcH: (1) low stiffness to measure the geometry of the object; (2) medium stiffness to match the estimated force necessary to deform the object when rolling it; and (3) high stiffness to compress the dough ball into a cookie.

First, the algorithm presented in Section 6.6 is used to estimate the force necessary to deform the piece of dough. StRETcH adjusts its stretch state to reflect the estimated stiffness given the current geometry of the dough formation. The UR5 proceeds to then roll the dough in a circle, checking the height of the object in parallel at 1Hz. When the standard deviation of the last ten observed object heights is less than 1mm, the robot stops rolling the piece of dough and assumes it is now a sphere. The high conformability of StRETcH enables tracking of the dough ball's height while providing enough stiffness to deform the dough during rolling. Furthermore, while traction resulting from tangential motions applied to the dough with large indentation depth is not modeled, these additional forces contribute in shaping the dough. Finally, StRETcH engages its highest stiffness by stretching to its maximum width and pressing down onto the dough to form a cookie. If the dough was successfully shaped as a sphere, the cookie would be round. The circularity, or $\frac{4\pi \text{area}}{\text{perimeter}^2}$ of each cookie was calculated, with 0.785 and 1 corresponding to a perfect square and circle, respectively. As shown in Figure 6.8, StRETcH successfully shaped the cookies (with average circularity of 0.888) from cubes of dough.

6.7 Conclusions

In this work, we presented a Soft to Resistive Elastic Tactile Hand called StRETcH. Modifying the design of the tactile sensor developed in [136, 73], StRETcH consists of an easily-manufactured membrane that is imaged by a depth sensor and uniaxially stretched by a parallel-jaw gripper. Contact geometry estimation using StRETcH can exhibit as low as sub-millimeter accuracy, and StRETcH can reconstruct the geometry of deformable objects such as paper cubes and wire shapes. The new actuation design enables controllable effective modulus from 4kPa to 9kPa. With a calibrated model of the mechanical response of StRETcH, errors in estimated contact forces are on average less than ± 0.025N. Furthermore, by modulating its stiffness, StRETcH was able to efficiently estimate a proxy stiffness for balloons filled with different materials. We concluded with a demonstration of StRETcH working in three different stiffness regimes to 1) estimate the geometry of a piece of dough, 2) deform the dough during rolling, and 3) press the dough ball into a cookie shape.

This work demonstrates new capabilities of variable stiffness soft tactile hands, and future work aims to improve on the design and modeling of StRETcH as well as to develop more efficient algorithms for manipulating solid deformable objects. First, the stiffness of StRETcH (and therefore the exerted force) is upper-bounded due to the thickness of the membrane and the maximum allowable stretch from the robotic gripper. Future designs of the sensor will focus on exploring new materials that have a larger range of stiffness when stretched (e.g., reinforcing the membrane with flexible fibers) and designing new mechanisms for uniaxial stretching that could miniaturize StRETcH onto an individual robotic finger for more dexterous in-hand manipulation. Furthermore, while the Hertz contact model approximation was shown to fit the experimental data quite closely, we fully acknowledge that a membrane is mechanically quite different and much more complex than an elastic half-space. Therefore, more physically-accurate modeling may be done by pairing state-of-the-art numerical simulation of membranes such as in [197] with a learning-based approach presented in [150] to estimate the contact force and stiffness of an object from contact geometry and the stretch state of the sensor. One could also imagine using a highly stretchable sensing skin such as in [105] to directly sense the internal strain of the membrane, which, paired with numerical simulation, could potentially make more accurate predictions of contact forces. Finally, StRETcH was able to roll a block of dough into roughly a sphere, but each roll-out required on average ten minutes, with some lasting up to twenty. In the following chapter, we explore using reinforcement learning techniques with StRETcH to efficiently shape dough into logs of varying lengths.

Chapter 7

Deformable Elasto-Plastic Object Shaping

This chapter is based on the paper "Deformable Elasto-Plastic Object Shaping with an Elastic Hand and Model-Based Reinforcement Learning" [130], written in collaboration with Ruzena Bajcsy.

Deformable solid objects such as clay or dough are prevalent in industrial and home environments. However, robotic manipulation of such objects has largely remained unexplored in literature due to the high complexity involved in representing and modeling their deformation. This chapter addresses the problem of shaping elasto-plastic dough by proposing to use a novel elastic end-effector to roll dough in a reinforcement learning framework. The transition model for the end-effector-to-dough interactions is learned from one hour of robot exploration, and doughs of different hydration levels are rolled out into varying lengths. Experimental results are encouraging, with the proposed framework accomplishing the task of rolling out dough into a specified length with 60% fewer actions than a heuristic method. Furthermore, we show that estimating stiffness using the soft end-effector can be used to effectively initialize models, improving robot performance by approximately 40% over incorrect model initialization. Experimental results can be viewed at `https://youtu.be/SKpmag1LlCY`.

7.1 Introduction

Robotic manipulation of deformable objects has applications in various domains, including robotic surgery, home robotic solutions, and automated food preparation. A ubiquitous challenge that persists throughout this area of robotics research is that robotic interactions with deformable material are highly complex, often nonlinear, and difficult to model. These complexities impose additional challenges of tractability in action selection for manipulating the deformable object. The particular task this work focuses on is shape-servoing elasto-plastic solid (3-dimensional) deformable objects. Specifically, the goal is to find a sequence of rolling actions to apply to a ball of dough to efficiently form it into a desired length. Rolling out dough into a log is a crucial part of many dough manipulation tasks, from dumpling making to bread shaping.

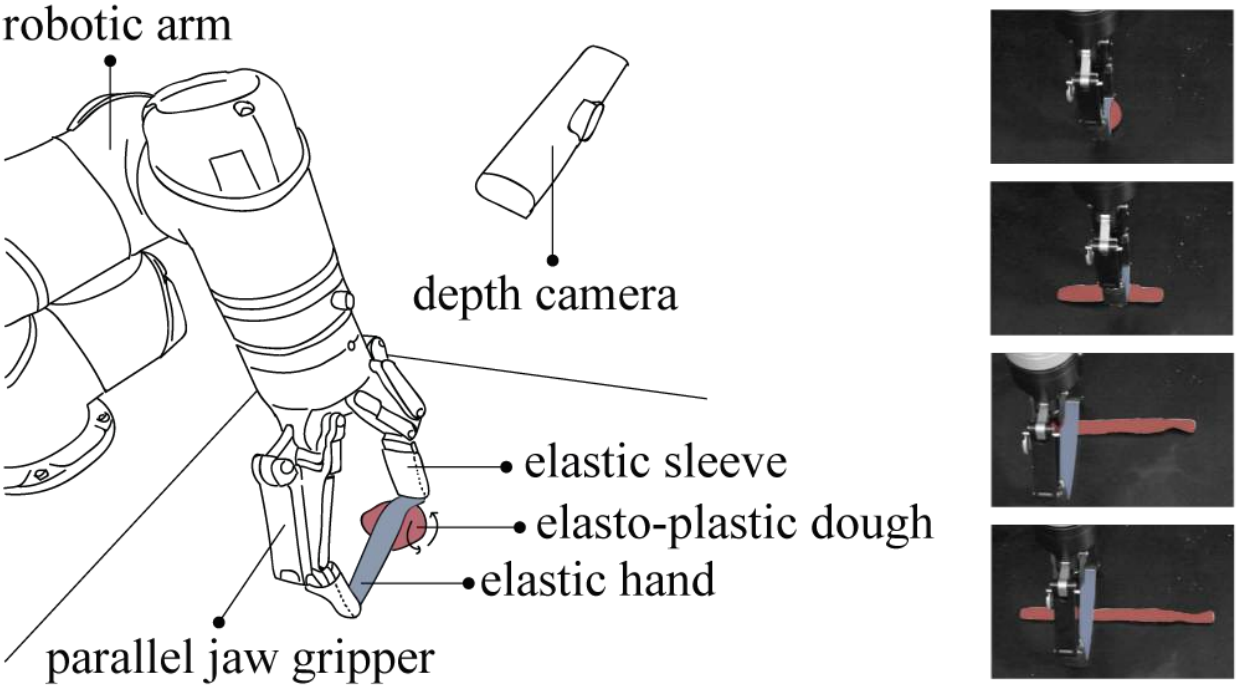


Figure 7.1: (left): Diagram of the robotic system. The elastic hand is used to roll the elasto-plastic dough back and forth. The gripper changes the hand's stiffness via stretching, which modulates the force imposed by the hand onto the dough. The overhead depth camera is used to track the progression of the dough's shape. (right): Four frames taken from a rollout of a dough ball into a 10 inch log. Image is colorized to match the diagram on the left. © 2021 IEEE

To address this task, we designed an elastic end-effector attachment for a parallel jaw gripper that can vary in stiffness via stretching. This design is a revision of the StRETcH sensor [131] presented in Chapter 6, a variable stiffness tactile end-effector. Changes were incorporated to compensate for kinematic limits of the robot arm and to allow for deeper contacts with the dough. Soft robotic end-effectors have low intrinsic stiffness and mass, making them particularly useful for handling delicate objects such as coral reefs [52] and wine glasses [2]. However, they can be challenging to manufacture or require additional tethering, cabling, and hardware. In contrast, the proposed elastic end-effector attachment is easy to assemble and can be made from two elastic bands. Furthermore, with the aid of a depth camera rigidly mounted above the work surface, it can be used to estimate the stiffness of objects with an active palpation technique proposed in the prior chapter.

Manipulation of clay or dough is typically non-prehensile, i.e., it involves pushing or moving the object against a surface [25]. In this work, we design a simple action space to roll out a piece of dough against a horizontal flat surface and track the dough's shape progression with a low-dimensional feature space. These simplifications help condense the complexity of the task, making model-based reinforcement learning a tractable solution. Dynamics models are learned for specific

dough through random exploration, and experiments demonstrate that model-based reinforcement learning is an effective way to complete the given task of dough rolling. Furthermore, we show that stiffness estimation can be used to match the correct dough model for more efficient rolling. To our best knowledge, this is the first work to propose a solution using a soft robotic end-effector to shape-servo a deformable elasto-plastic object.

The main contributions of this chapter are:

- A new elastic end-effector attachment designed for shaping solid deformable objects that is cost-effective and easy to integrate with a parallel-jaw gripper. The elastic end-effector is also used to estimate the stiffness of different dough, which improves rolling performance.
- An algorithm leveraging model-based reinforcement learning and simple featurization and actions that enable tractable optimization for the real-time solid deformable object manipulation task.
- Robotic experiments that roll doughs of different hydration levels into varying lengths. These experiments serve to quantify and demonstrate the efficacy of the proposed hardware and software framework.

7.2 Related Work

Deformable Object Manipulation

Commonplace in many domains like the home environment, deformable objects are categorized as linear, planar, and solid objects [177]. Efforts in robotic manipulation of deformable objects have largely focused on manipulating linear objects such as rope [214, 195] and cables [231], as well as planar objects like paper [8], cloth [139, 129] and gauze [199]. This particular work is interested in studying the manipulation of solid deformable objects like dough, which remains the least researched category within robotic deformable object manipulation [177].

The complexities involved in representing a solid deformable object's shape as well as modeling its interactions with a robot and the environment contribute to the difficulty of this area of research. Thus, there has been significant effort in designing effective ways to parameterize deformable objects. For instance, a variety of approaches have been proposed for tracking the contour or shape of a deformable object during manipulation, ranging from continuum mechanics and simulation-based methods [164, 55] to neural network-based methods for contour segmentation and deformation tracking [35, 34, 188]. However, due to the nature of this work's defined task and its objective, simple geometric features are demonstrated to be a sufficient parameterization of the dough.

In terms of designing control or action selection policies to effectively manipulate deformable solid objects, prior work has utilized both model-based [65, 219] and model-free adaptive methods [153, 25]. While simulation-based methods have been shown to be quite useful in the space of linear and planar object manipulation [195, 53], it is more computationally expensive to collect

large amounts of data for physically-realistic simulations of dough. Learning from demonstrations has therefore been cited to be the most popular method for training a robot to shape deformable objects [147], accomplishing tasks such as shaping sand [25] and pizza dough [51]. Similar to the work in [65, 219], this work uses a model-based method, where material-relevant parameters are estimated and then used to guide the control strategy. However, the model does not explicitly represent the deformations of the object but rather captures its transitions resulting from robot interactions. Because this work is model-based, it requires significantly less data than model-free methods.

This work is unique in that it is the first work to propose using a soft robotic end-effector as the tool for shape-servoing dough. Furthermore, intentional design of simple feature and action spaces allows for real-time model-based reinforcement learning to solve the desired task.

Soft Robotic End-effectors

A soft robotic end-effector was designed specifically for the task of rolling and shaping dough. Composed entirely of an elastic band, the end-effector's variable stiffness allows it to be sensitive to highly deformable objects while capable of imposing enough force to exert work on the object in order to change its shape. Soft robotic end-effectors have been used for grasping objects as delicate as coral reef [52], maneuvering fragile wine glasses [2], and supporting and interacting with humans [71]. The ability to conform to the environment and controllably vary in stiffness is crucial when handling soft, elasto-plastic objects like dough, which will permanently deform under large forces or flatten from rigid end-effectors.

Soft robotic end-effectors have also been coupled with tactile sensing. Capable of providing high-resolution images of contact area, soft tactile sensors have been used to detect tumors in tissue [206] as well as estimate the hardness of fruits [224].The proposed end-effector in this chapter is a variation of the Soft to Resistive Elastic Tactile Hand (StRETcH) [131], which is a variable stiffness tactile end-effector made of elastic material. In Chapter 6, StRETcH was used to measure both the stiffness of different objects as well as their geometry under contact. Preliminary experiments showed promising results in using variable stiffness to roll a block of Play-Doh into a sphere via a heuristic circular motion. We hope to expand on that concept by pairing the new iteration of StRETcH with an action-selection algorithm to roll a ball of Play-Doh into logs of different lengths.

Model-Based Reinforcement Learning

The proposed action-selection algorithm for this work is rooted in model-based reinforcement learning. Reinforcement learning (RL) has been used to manipulate linear and planar objects using both model-free [117, 129, 78] and model-based approaches [62, 46]. Much of this prior work relies on simulation in order to collect training data, as physical experiments are expensive. However, simulated representations of three-dimensional deformable objects like clay or dough are complex or physically inaccurate and thus have not been substantially explored. Model-free RL has been shown to be effective at learning control policies directly from high-dimensional input,

although these methods require a considerable amount of data, while model-based RL is much more sample efficient at the cost of scaling poorly with high-dimensional tasks [122]. This chapter largely takes inspiration from the model-based RL framework proposed in Nagabandi et. al. [149], which highlights the sample efficiency achieved with a model-based approach. Furthermore, the parameterization of observations and simplification of the action space ensures that this task is low-dimensional and therefore optimization is tractable. Data collection for training the transition model of the nonlinear end-effector-to-dough interactions is simulation-free and only uses approximately one hour of real robot time per dough. Experiments demonstrate that model-based RL efficiently accomplishes the task of dough rolling, even for poorly initialized models.

7.3 Problem Statement

Definitions and Objective

The objective of this chapter is to design a policy to select a sequence of rolling actions to apply to a ball of dough, using the elastic end-effector, to form it into a desired length. More specifically, at time step t, the dough will be in a state $\mathbf{s}_t \in \mathcal{S}$. In reality, $\mathbf{s}_t$ will represent a partial observation of the true state of the dough. Using the elastic end-effector, the robot can apply a rolling action $\mathbf{a}_t \in \mathcal{A}$ to the dough. Upon execution of the action, the dough will transition to a new state $\mathbf{s}_{t+1}$. This transition is assumed to be captured by an underlying dynamics or transition function $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$.

Since the goal is to find a sequence of actions that will form the dough into the desired shape in the least amount of steps, the problem can be naturally formulated in a reinforcement learning (RL) framework. In RL, a reward $r_t = r(\mathbf{s}_t, \mathbf{a}_t)$ is received at each time step t, and the robot or agent is tasked with taking the action that will maximize the cumulative reward over future time steps. We design the reward function to directly reflect task progression, i.e., the shape of the dough at time t ($\mathbf{s}_t$) relative to the goal state $\mathbf{s}_g$. Thus, an H-length sequence of future actions can be selected by evaluating the objective function at each time step t:

$$(\mathbf{a}_t, \ldots, \mathbf{a}_{t+H-1}) = \underset{\mathbf{a}_t, \ldots, \mathbf{a}_{t+H-1}}{\arg\max} \sum_{n=0}^{H-1} \gamma^n r(\mathbf{s}_{t+n}, \mathbf{a}_{t+n}) \tag{7.1}$$

where $\gamma \in [0, 1]$ is a discount factor that places more weight on more immediate rewards.

Assumptions

The interaction between an elastic membrane and an elasto-plastic object is complex and difficult to model, but the following assumptions allow for a tractable, real-time solution. First, when the robot is not in contact with the dough, we assume that the dough's state is in equilibrium and unchanging. This allows for the simplification of transition dynamics, where each state observation $\mathbf{s}_t$ is taken at a time step t when the robot is no longer in contact with the dough. The transition function

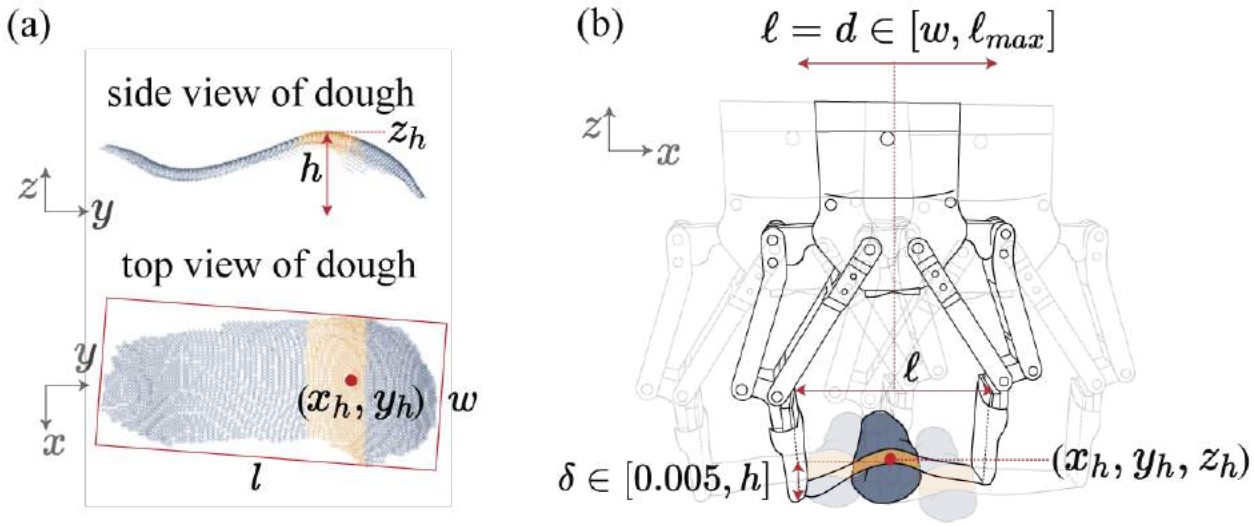


Figure 7.2: (a): Featurization of a point cloud of dough. The state observation $[l, w, h]$ is extracted by finding the minimum bounding box enclosing this point cloud. The yellow region signifies where the robot will contact the dough in the next roll. (b): Each roll is state-dependent. The length of the elastic band ℓ and the distance of the roll d are fixed to be equivalent and lower-bounded by w. The depth of the indentation δ is upper-bounded by h. © 2021 IEEE

f models the change in state due to this applied action and thus is not time-dependent. Rather, the problem can now be considered in a discrete-time formulation, where samples are observed between what are assumed to be stationary interactions. Additional assumptions include that the dough is uniform, it is not sticky or softer than the elastic membrane when un-stretched, and there exists enough friction between the dough and the work surface so that the dough will roll instead of slide under applied forces from the end-effector. Furthermore, the goal states are assumed to be reachable from the initial spherical state of the dough. With these assumptions, explicitly modeling the complex interactions between the dough and the soft elastic end-effector can be circumvented. Instead, the transition function f summarizes the bulk effects due to the applied actions at a sparse timescale that is sufficiently descriptive and computationally friendly.

7.4 Methods

Featurization of Dough

Due to the objective's geometric nature, a low-dimensional feature vector was used to capture the shape of the dough. The chosen state space is defined as $S = \{l, w, h, x_h, y_h\}$, a five-dimensional space composed of the length, width, and height (in meters) of the minimum bounding box enclosing the dough, and the $x-y$ position of the highest point of the dough (x_h, y_h, z_h). The goal

states are uniform cylinders of varying lengths, so this low-dimensional feature space is sufficient for the task. For more complicated objectives, e.g., forming a shape that cannot be parameterized by simple geometric features, a larger state space can be explored.

To extract these features from the original point cloud of the dough, a depth image of the workspace is sampled when the end-effector is not in contact with the dough. The depth image is projected into a 3D point cloud using camera intrinsics, and the point cloud is transformed into world coordinates, with the origin centered at the base of the robot. The workspace surface and the robot end-effector are filtered from the transformed point cloud, resulting in a point cloud of the surface of the dough. The highest point (x_h, y_h, z_h) is determined from the filtered point cloud. h is calculated by subtracting the maximum height (z_h) of the point cloud from the height of the workspace surface, and l and w are calculated by finding the minimum-area bounding rectangle of the points, projected into the $x-y$ plane. Feature extraction is illustrated for an example point cloud in Figure 7.2a.

Simplified Action Space

While we acknowledge the diversity of actions that can achieve the desired goal states, we greatly simplify the search problem by reducing the action space to five dimensions. To do this, we constrain most of the rolling motion applied at each step t. Furthermore, added heuristic constraints compress the effective search space to two dimensions.

First, let a rolling motion $m(\ell, d, x, y, z)$ be defined by the length of the elastic band ℓ (i.e., the gripper throw), the distance of the roll d, and the position (x, y, z) of the robot hand where the roll begins and ends. Each rolling motion is constrained to move in a back and forth motion along the direction of the x-axis, which should lie approximately perpendicular to the length of the dough formation. Thus, the entire roll begins with the robot end-effector positioned at (x, y, z), moves to position $(x + \alpha * \frac{d}{2}, y, z)$, position $(x - \alpha * \frac{d}{2}, y, z)$, and ends at position (x, y, z), where α is randomly assigned 1 or -1 each time t for symmetry.

To further simplify the search space over actions $\mathcal{A} \in \{\ell, d, x, y, z\}$, a heuristic policy targets every roll to begin at the $x-y$ position of the highest point of the dough formation $(x_h, y_h, z_h)_t$. That is, the roll motion fixes $x = x_h$ and $y = y_h$. Roll distance d is also constrained to equal the parallel jaw gripper throw ℓ. These heuristic constraints keep the dough fairly even during rollouts and reduce the effective action space. Thus, the two remaining variables for the rolling motion m are ℓ and z. We lower bound the range of ℓ at time t by the width of the dough w at that time step. Varying ℓ subsequently varies the stiffness of the elastic band, which consequently changes the force applied to the dough during rolling contact. Finally, the vertical depth at which the elastic band presses into the dough can be calculated by $\delta = z_h - z$. We impose the following bounds to the depth $\delta \in [0.005, h]$ to guarantee contact with the dough for all rolls while ensuring that the robot does not violate physical boundaries. The effective two-dimensional action search space used to parameterize the rolling motion thus consists of the length of the elastic band ℓ and the pressing depth $\delta = z_h - z$. These are summarized in the diagram in Figure 7.2b. Note that all roll motions are executed at the same speed.

Random Exploration

Random interactions with the dough are used for two purposes – to collect data to learn a dynamics function f and to use as comparison against the proposed action-selection algorithms. At time step t, a random action $\mathbf{a}_t = (\ell_t, \delta_t)$ is uniformly sampled from a state-dependent distribution. Specifically, the length of the elastic band ℓ_t is sampled from a uniform distribution bounded by $[w_t, \ell_{max}]$, where w_t is the current width of the dough and ℓ_{max} is the maximum gripper throw. The pressing depth δ_t is sampled from a uniform distribution bounded by $[0.005, h_t]$. Once the random action a_t is sampled, a rolling motion $m(\ell_t, \ell_t, x_h, y_h, z_h - \delta_t)$ is executed, i.e., the gripper is commanded to stretch the elastic band to length ℓ_t, the distance of the roll is ℓ_t, and the start and end position of the roll is defined by $(x_h, y_h, z_h - \delta_t)$.

Random exploration thus consists of a sequence of actions sampled from the updating state-dependent bounds for an indeterminate amount of time for off-policy learning or until the goal state is reached. We determine that the goal state is reached if the Euclidean distance from the current state $\mathbf{s}_t$ to the desired state $\mathbf{s}_g$ is less than a pre-defined ε.

Heuristic Rolling

For the proposed heuristic rolling method, each roll is still applied at the position of the highest point of the dough formation (x_h, y_h), but a fixed policy is used for action selection. ℓ_t is fixed at $\frac{4}{5}\ell_{max}$ for high force output, and δ_t is fixed at $z_h - (z_g/2)$. At first, δ_t was fixed at $z_h - z_g$, or the difference in height of the current state to the goal state. However, this proved to be an inefficient heuristic as the dough state progressed toward the goal, since δ_t would gradually decrease and subsequently the imposed force would decrease as well. We therefore chose to fix δ_t at $z_h - (z_g/2)$ for greater force output throughout the rollout, although any choice of ℓ_t and δ_t is arbitrary without explicit knowledge of the hand-dough dynamics. These fixed actions are applied until the goal state is reached (the Euclidean distance from the current state to the desired state is less than ε).

Dynamics Model

Training data for the dynamics model is collected via random exploration, resulting in a series of trajectories $(\mathbf{s}_0, \mathbf{a}_0, \ldots, \mathbf{s}_{N-1}, \mathbf{a}_{N-1}, \mathbf{s}_N)$, each of which has N transitions. Training set $\mathcal{D}_{off}$ consists of input state-action pairs $\{(\mathbf{s}_t, \mathbf{a}_t)\}$ with a corresponding output set of $\{\mathbf{s}_{t+1} - \mathbf{s}_t\}$. Note that only a subspace of the state space ($\{\ell, w, h\}$) and action space ($\{\ell, \delta\}$) is used to learn the dynamics model, since (x_h, y_h) in early experiments did not demonstrate significant effect on the subsequent transition, while the remaining action space was heuristically constrained. The dynamics model $\hat{f}_\theta(\mathbf{s}_t, \mathbf{a}_t)$, parameterized by weights θ, can be learned via multiple machine learning techniques. Given the low-dimensional state and action spaces, we chose to represent $\hat{f}_\theta$ with a linear regression model with polynomial features of degree 2.

Rolling with Model-Based Reinforcement Learning

In order to perform action-selection using model-based reinforcement learning, we largely take inspiration from recent advances in model-based RL robotics research [208, 149, 102, 148]. These works use Model Predictive Control (MPC) to select the next action to execute within a Reinforcement Learning framework to iteratively improve the dynamics model. Model Predictive Control solves the objective defined by Eqn. 7.1 at each time step t, executes the first action in the optimal sequence a_t, and repeats this process at the next time step once state s_{t+1} is observed. MPC assumes knowledge of system dynamics, so the executed action sequence may be suboptimal given an incorrect or poorly-initialized model. To compensate for these errors, the reinforcement learning framework iteratively refines the dynamics model by refitting the weights of $\hat{f}_\theta$ every T steps, using both the original data $\mathcal{D}_{off}$ along with the on-policy data collected during execution $\mathcal{D}_{on}$.

Iterative Random Shooting with Refinement (Cross Entropy Method)

MPC for nonlinear systems is an active area of research, and there are several methods that can be used to optimize the objective function in Equation 7.1. One such method that is considered for this work is iterative random shooting with refinement, which is a cross-entropy method (CEM) approach [17]. Random-sampling shooting samples K candidate action sequences of length H for a planning horizon of H steps and selects the action sequence corresponding to the highest cumulative reward ($\sum_{n=0}^{H-1} \gamma^n r(\mathbf{s}_{t+n}, \mathbf{a}_{t+n})$), where future states $\mathbf{s}_{t+n}$ for $n \in [1, H)$ are predicted via the trained dynamics model $\hat{f}_\theta(\mathbf{s}_{t+n-1}, \mathbf{a}_{t+n-1})$. The CEM approach improves random-sampling by iteratively refining the mean and variance of the sampling distribution for M iterations, taking the mean of the last distribution as the optimal action sequence. In other words, for each iteration, random-sampling shooting generates K candidate action sequences by sampling:

$$a_t^k \sim \mathcal{N}(\mu_t^m, \Sigma_t^m), \forall k \in K, t \in [0, H), m \in [0, M)$$

After K sequences are sampled, the top N sequences (A_{top}) corresponding to the highest cumulative rewards are used to refine the mean and variance of the next iteration:

$$\mu_t^{m+1} = \beta \text{mean}(A_{top}) + (1-\beta)\mu_t^m, \forall t \in [0, H)$$

$$\Sigma_t^{m+1} = \beta \text{var}(A_{top}) + (1-\beta)\Sigma_t^m, \forall t \in [0, H)$$

where β is used to smooth the updates. After M iterations, $\mu_t^M \forall t \in [0, H)$ is chosen as the optimal action sequence.

Numerical Optimization with Powell's Method

An alternative gradient-free approach is to maximize the objective in Equation 7.1 using a numerical technique such as Powell's conjugate direction method. Given the low-dimensional action and state space, for a short-term planning horizon, this remains a tractable solution. We use Scipy's minimize function implementation of Powell's method to solve for the optimal sequence $(\mathbf{a}_t, \ldots, \mathbf{a}_{t+H-1})$, given Equation 7.1 and the transition function $\hat{f}_\theta(\mathbf{s}_t, \mathbf{a}_t)$ to estimate future states.

Model Matching via Stiffness Estimation

In Chapter 6, we demonstrated how a proxy for the stiffness of deformable objects can be estimated by palpating the objects from above and mapping the indentation depth and stretch state of the elastic hand to a predicted output force F [131]. That is, we fit a model $F = h(\ell, \delta)$, and a one-dimensional measure for stiffness is calculated by dividing F by the object's z deflection. As discussed in Chapter 6, $h(\ell, \delta)$ is also dependent on contact geometry, but given that stiffness estimation is used only for initializing the model, and the dough is assumed to be spherical with a known diameter, this work does not consider the effect of contact geometry when estimating the imposed force. A reasonable estimate of force can be made by first measuring the varying force while modulating the indentation depth and stretch state of the elastic hand against a rigid, hemispherical probe (of equivalent diameter as the dough) attached to an ATI Axia80 EtherNet Force/Torque sensor. These force measurements are depicted in Figure 7.3 and the model $F = h(\ell, \delta)$ is fit to this data. As shown in Figure 7.3, force output increases with increasing indentation depth and stretch. When estimating the stiffness of an object, the robot actively palpates the object until a deflection is observed by the overhead camera. At the point of deflection, the predicted force is divided by the amount of deflection, giving a proxy estimate of stiffness in N/mm.

We can use these proxy estimates of stiffness to initialize dynamics models for unknown dough. The stiffness of a dough is dependent on its hydration level, or its water content, and stiffness directly affects the material properties and subsequently the nonlinear dynamics of the dough. In Section 7.5, we discuss how we design a soft dough, a stiff dough, and a medium-stiff dough by varying their hydration levels. We train a separate model for the dry and hydrated dough. However, to estimate the dynamics of an unknown dough, e.g. the medium-stiffness dough, we can linearly interpolate the predictions from the two given models by weighting their predictions based on stiffness. That is, let the stiffness and model of the dry and hydrated dough be defined as σ_{dry}, $\hat{f}_{dry}(\cdot)$, $\sigma_{hydrated}$, and $\hat{f}_{hydrated}(\cdot)$. If the estimated stiffness of the unknown dough is $\sigma_{unknown}$, then the corresponding dynamics model for the new dough is:

$$\mathbf{s}_{t+1} = (1-\beta) * \hat{f}_{hydrated}(\mathbf{s}_t, \mathbf{a}_t) + \beta * \hat{f}_{dry}(\mathbf{s}_t, \mathbf{a}_t) \tag{7.2}$$

where $\beta = \frac{\sigma_{unknown} - \sigma_{hydrated}}{\sigma_{dry} - \sigma_{hydrated}}$ and $\sigma_{unknown}$ is bounded by $[\sigma_{hydrated}, \sigma_{dry}]$. Stiffness is likely not simply linearly related to the resulting dynamics of the dough, but later experimental sections show that initializing the dynamics model by this method still improves performance over incorrectly initializing it to either $\hat{f}_{hydrated}(\cdot)$ or $\hat{f}_{dry}(\cdot)$.

7.5 Experimental Setup

Elastic Hand Design

The design of the elastic hand is a variation of Chapter 6's StRETcH [131], a soft tactile hand attached to a parallel jaw gripper that varies in stiffness via opening and closing the gripper. In StRETcH, tactile images are observed via an overhead camera mounted rigidly to the robot arm.

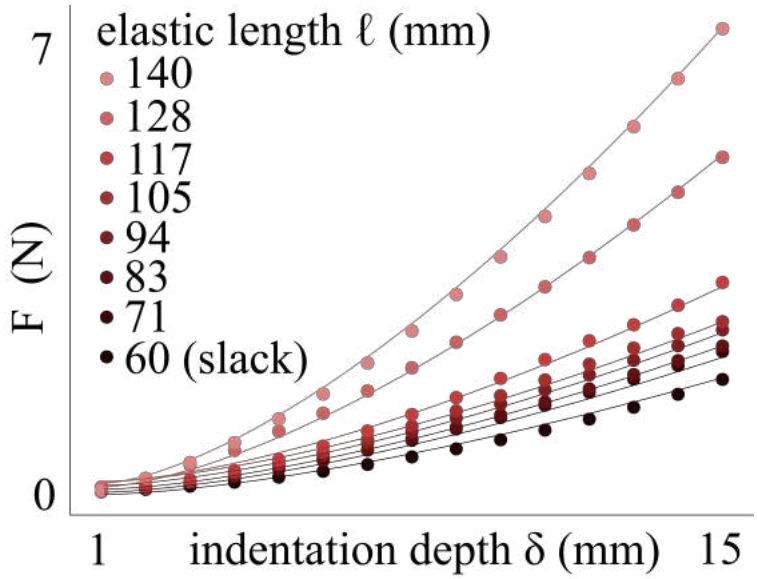


Figure 7.3: Indentation depth δ vs. load force F for varying stretch states for a 30mm diameter hemispherical rigid indenter. © 2021 IEEE

However, the configuration of the elastic membrane along the length of the robot fingers was kinematically limiting for this task. Thus, for this chapter, the StRETcH sensor was redesigned with the elastic surface rotated by 90 degrees (see Figures 7.1 and 7.2). Furthermore, the only materials used to construct this new end-effector attachment were two elastic bands. The stretchier band was used as the soft contact interface for the dough. The stiffer elastic material was sewn into two separate loops, which were sewn to each end of the stretchier band. The stiff loops acted as sleeves that were securely attached to the robot fingers via friction fit, positioning the softer band between the fingers. This attachment was cost-effective to make and required no additional hardware or cabling.

Physical Setup

The elastic end-effector attachment was secured via friction fit to a Robotiq 2-Finger Adaptive Robot Gripper, which was attached to a Universal Robot UR5 arm. In contrast to the experimental setup in Chapter 6, in which the camera was mounted rigidly to the robotic arm, this chapter fixes the Intel RealSense Depth Camera SR305 at a stationary position, looking down at a slight angle onto the workspace surface. The reachable workspace surface was approximately 1ft$\times$1ft.

Dough Consistency

Play-Doh was used as the elasto-plastic material for all experiments. Some of the following experiments explored rolling dough of varying stiffnesses. To create three different stiffnesses from

the same Play-Doh, each 50 gram portion of Play-Doh was hydrated at a different hydration level. To hydrate the dough, different amounts of water were kneaded into the dough until the water was completely absorbed. The more water incorporated, the softer the dough became.

Using the method developed in Chapter 6 and described in Section 7.4, the robot measured the stiffness of 50 grams of dough when hydrated with 1, 2, 3, 4, and 5 grams of water. The approximate stiffnesses were 1.23, 0.97, 0.85, 0.67, and 0.49 N/mm, respectively. Thus, the three doughs used for experiments in Sections 7.6 and 7.6 were hydrated with approximately 1, 3, and 5 grams of water. Beyond 6 grams, the dough became too sticky to shape using the elastic membrane. The rest of this chapter refers to the doughs as:

dough	A	B	C
water (g)	5	3	1
stiffness (N/mm)	0.49	0.85	1.23

Table 7.1: Hydration levels and proxy stiffness values of doughs A, B, and C.

Data Collection

We collected a total of two hours of training data, with one hour each for dough A and C. Each initialization began with the dough hand-rolled into a roughly spherical shape. The dataset for dough A consisted of four initializations, with 150 random actions sampled for each. For dough C, the training trajectories consisted of six initializations, with 100 random actions sampled for each. Random exploration was restarted with a new initialization once rate of progress (dough-lengthening) appeared to diminish substantially, which accounts for the difference in initializations between dough A and C. Each dough dataset had a total of 600 actions and transitions, and a separate model was trained for each dataset.

7.6 Results

Comparison of Rolling Algorithms

We begin by comparing the four proposed action-selection methods: random sampling, a heuristic method with a fixed action policy, RL with random shooting and CEM, and RL using numerical optimization. Both RL algorithms use MPC for action selection and iteratively refine the dynamics model as the robot experiences new data. For both RL methods, the planning horizon (H) and model iteration horizon (T) were both chosen to be 10 steps, and the CEM method additionally used $K = 100$, $M = 3$, $\beta = 0.5$ as hyperparameters. New data $\mathcal{D}_{on}$ were weighted $10\times$ more than $\mathcal{D}_{off}$ when re-fitting $\hat{f}_\theta(\cdot)$. For this experiment, all rolling was performed on dough A, and the dynamics models were initialized to $\hat{f}_A$. For each method, six independent rollouts were run for five different goal lengths of 2, 4, 6, 8, and 10 inches, and the number of steps until the goal was reached was recorded.

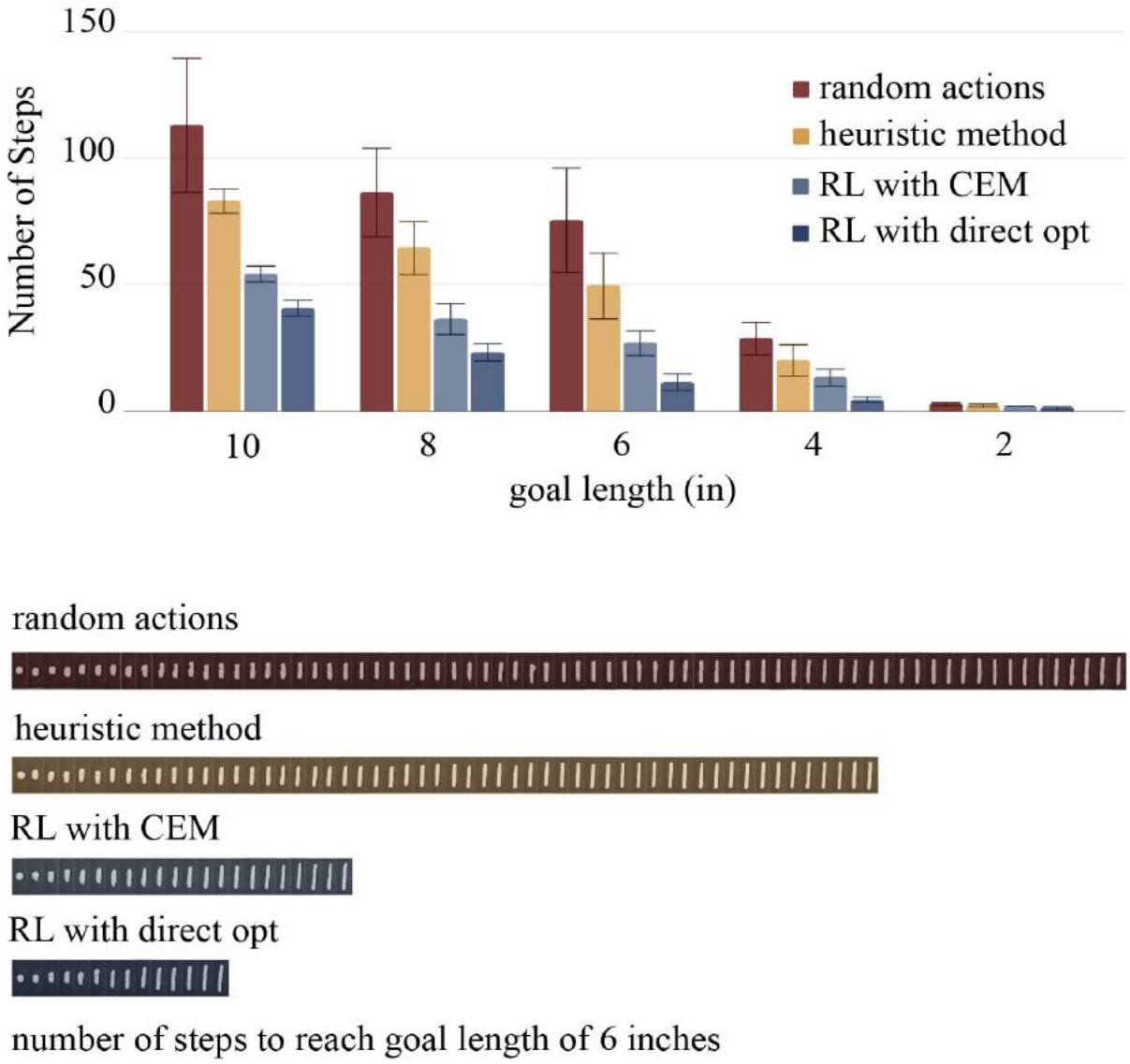


Figure 7.4: (top) Number of steps to roll a ball of dough into the goal length for four different action-selection methods. 6 trials are represented per bar and error bars reflect a 95% confidence interval. (bottom) An example of the dough state progression at each time step for all four methods. © 2021 IEEE

Figure 7.4 summarizes the results of the experiment. The heuristic method was on average $27.6 \pm 6.1\%$ more efficient than purely random sampling, and both RL methods were more efficient than the heuristic method ($33.4 \pm 18\%$ and $63.1 \pm 18.3\%$, respectively). Numerical optimization was more efficient than RL with CEM across all goals, so the following experiments used numerical optimization.

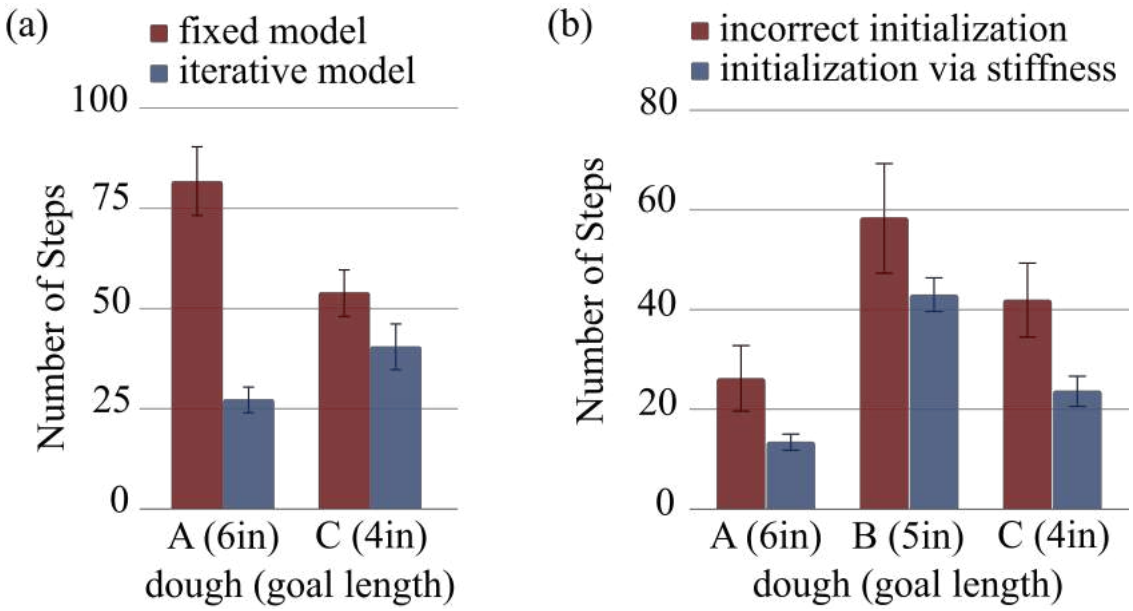


Figure 7.5: (a): Comparison of using a fixed vs. iterative model for Model Predictive Control. The iterative model is refined every $T = 5$ steps from new data collected at run-time. In both scenarios, the initial model is incorrect (dough A is initialized with $\hat{f}_C$ and vice versa). The goal states for doughs A and C are 6*in* and 4*in* in length, respectively. 15 trials are represented per bar. (b): Comparison of incorrectly initializing the model vs. initializing the model using the estimated stiffness of the dough. Dough A is incorrectly initialized with $\hat{f}_C$, Dough C is incorrectly initialized with $\hat{f}_A$, and Dough B is initialized by either $\hat{f}_A$ or $\hat{f}_C$ with equal probability. The goal states for Doughs A, B, and C are 6*in*, 5*in*, and 4*in* in length, respectively. 10 trials are represented per bar.

Comparison of Fixed versus Iterative Models

In the first experiment, the dynamics model was initialized correctly for the corresponding dough, and model-based RL methods achieved the goal state efficiently. In this experiment, we tested the robustness of the RL framework to iteratively correct a poorly-initialized or incorrect dynamics model. To do so, we compared using MPC to roll out dough using a fixed dynamics model versus an iteratively improved model using RL. Furthermore, for both cases, we incorrectly initialized the dynamics model – dough A trials were initialized with $\hat{f}_C$, and similarly dough C trials were initialized with $\hat{f}_A$. This experiment aimed to test the self-correcting ability of the RL framework. The goal states of dough A and C were set to 6in and 4in in length, respectively. This difference was to compensate for the fact that dough C was stiffer and thus took longer to roll out than dough A. For this experiment, numerical optimization directly solved Equation 7.1 with $H = 5$ as the planning horizon at each step and $T = 5$ steps were taken before each model iteration. As in the previous experiment, new data $\mathcal{D}_{on}$ were weighted $10\times$ more than $\mathcal{D}_{off}$ when re-fitting $\hat{f}_\theta(\cdot)$. For

both fixed and iterative models, 15 rollouts were run per dough.

As shown in Figure 7.5(a), the iterative model outperformed the fixed dynamics model using MPC. Using RL to iteratively refine the dynamics model reduced the number of steps to reach the goal by, on average, 45.8%. Furthermore, it was originally expected that MPC initialized by dough C would select more aggressive (high force output) actions to compensate for the higher stiffness and thus the fixed model would outperform the iterative model for dough A. However, as dough A's state got closer to 6*in*, we observed actions by the robot that were less and less effective at inducing change in the dough's state. This was because the datapoints of dough C were very sparse in that goal range due to the nature of the stiffer dough and limited training time. However, by allowing model iteration via the RL framework, the model was quickly updated to better reflect the dough's true dynamics and the goal was reached in a short amount of time steps.

Figure 7.6 illustrates the average actions selected throughout the task, where the x-axis is the percent of total steps taken before the goal was reached. Comparing the fixed (red) and iterative (yellow) trends, one can see how the iterative model corrects throughout the progression of the task, although with considerably higher variance. For dough A, while the fixed model has the robot repeatedly performing a shallow roll with low ℓ, the iterative model pulls the action space towards higher values for ℓ. For dough C, the iterative model applied either high ℓ and low δ actions or low ℓ and maximum δ actions, resulting in high variance between the different rollouts.

Stiffness Estimation for Model Matching

It is likely that robots will not have models for every dough they will need to manipulate. We have seen that the RL framework can compensate for an incorrect or poorly-trained model by iteratively refining the dynamics model. However, we hypothesized that estimating stiffness could improve model initialization and therefore execute the task more efficiently than initializing the dynamics model to a random dough. This was a reasonable proposition, since stiffness reflects material properties relevant to dough dynamics.

To test this hypothesis, we compared using model-based RL (using numerical optimization) with incorrect initialization versus initialization via stiffness estimation. For each initialization method, 10 rollouts for doughs A, B, and C were run, with goal states of 6, 5, and 4in, respectively. For incorrect initialization, dough A was initialized with $\hat{f}_C$, and dough C was initialized with $\hat{f}_A$. For dough B, for which there was no learned model, 5 of its rollouts were initialized with $\hat{f}_A$, and the other 5 were initialized with $\hat{f}_C$. For the initialization method that incorporated estimated stiffness, the dynamics model was initialized by Equation 7.2. This experiment used the same hyperparameters as in Section 7.6.

As shown in Figure 7.5(b), initializing the models based on stiffness was significantly more efficient than initializing the model incorrectly. On average, initializing the model via stiffness estimations improved performance by 39.6%. Dough B took more steps to reach its goal than doughs A and C, which may suggest that the interpolation method can be optimized to better reflect the relationship between stiffness and dough dynamics (i.e., this relationship is nonlinear).

Figure 7.6 compares the incorrectly initialized (yellow) and the stiffness-initialized (blue) models. The actions selected throughout the rollouts have considerably lower variance when the model

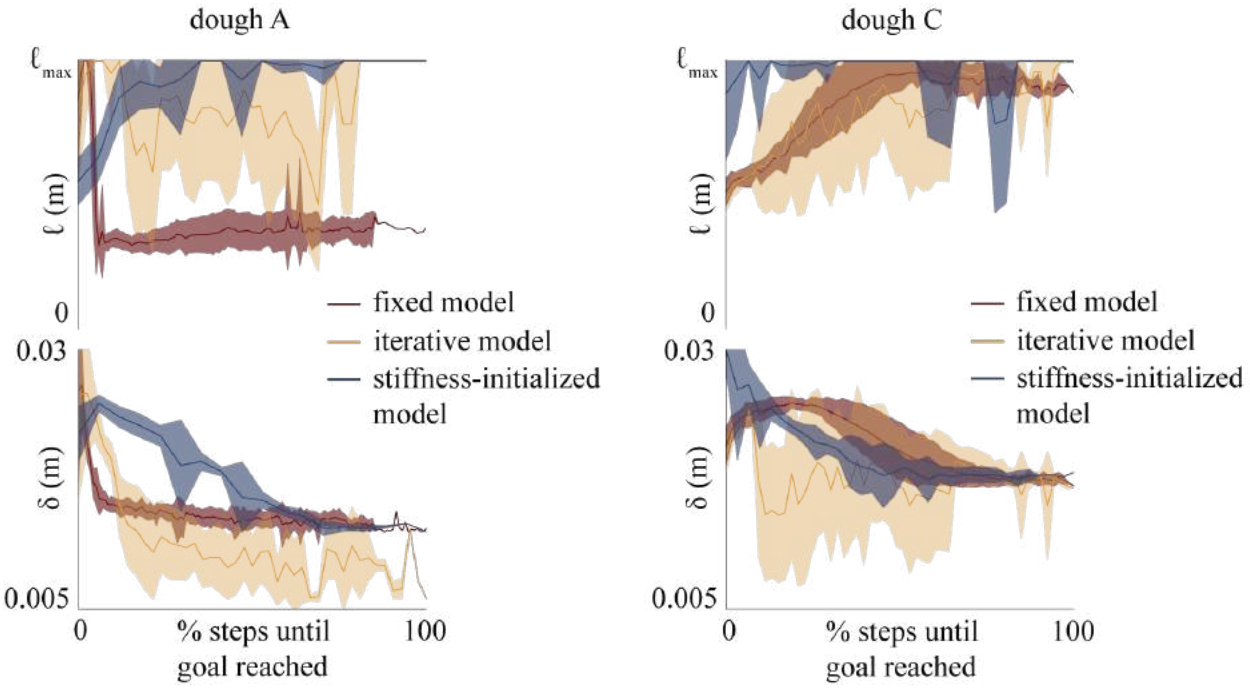


Figure 7.6: (left): Comparison of actions chosen throughout the progression of the task for dough A, with red, yellow, and blue corresponding to the fixed incorrect dynamics model of $\hat{f}_C$, iterative dynamics model initialized at $\hat{f}_C$, and stiffness-initialized model, respectively. The top graph is the gripper throw (length of the elastic band, ℓ) and the bottom graph is rolling depth δ. (right): Comparison of actions for dough C, where the incorrect initialized dynamics model is $\hat{f}_A$. © 2021 IEEE

is initialized by stiffness. The policy that seems to work effectively for dough A is initially rolling with low ℓ and high δ, and gradually increasing ℓ throughout the rollout. For dough C, the selected action sequences tended to prefer high ℓ and δ for high force-output, which is consistent with dough C's higher stiffness. Because many of the actions tended to prefer higher stiffness of the end-effector, a future consideration is to explore higher-stiffness elastic bands, or bands with greater ranges of variable stiffness.

7.7 Conclusions

To the best of our knowledge, this is the first work that accomplishes deformable manipulation using a soft end-effector on elasto-plastic dough. The novel end-effector is not only cost-effective and easy to integrate with a parallel-jaw gripper, but it also can vary in stiffness, which allows it to effectively manipulate soft deformable objects like dough. A model-based RL framework is leveraged to roll dough into different lengths, where simple featurization and actions enable

tractable, real-time optimization. The robot is capable of accomplishing the task after only one hour of random exploration, and the RL framework iteratively improves the transition model to compensate for incorrect or poorly-defined models. Real robot experiments demonstrate high efficacy of the proposed system, with RL action-selection exceeding the heuristic method by more than 60% in efficiency. Furthermore, stiffness estimation using the soft end-effector enables better model initialization, which consequently improves the robot's performance by approximately 40%.

We hope the experimental results of this chapter can inspire more efforts in tackling deformable solid object manipulation. Although the interactions involved in deformable object manipulation are highly complex and difficult to model, task-specific design of the state and action spaces can enable the robotic task to be much more tractable. In future work, other low-dimensional parameterizations of dough can be explored, including neural network-based encoders that may facilitate in tasks requiring the manipulation of dough into more complex geometries and shapes. Furthermore, more model representations should be experimentally compared, including Gaussian Process Regression and Neural Network Dynamic Models. Ultimately, we hope these findings can be used to manipulate real dough into complex shapes such as pretzels and challah bread, which all require rolling dough into logs.

Part IV

Discussion and Future Work

Chapter 8

Towards Dexterous Manipulation of Liquids, Grains, and Doughs

This thesis develops new techniques in interactive perception to learn physical properties of liquids, grains, and doughs, materials commonly found in environments like the kitchen but which often present challenges for existing robotic systems. Through the use of non-traditional sensory modalities such as sound and soft tactile sensing, the robot estimates material properties such as the coefficient of restitution and a proxy for stiffness. As these measurements directly relate to the dynamics of the material, the robot is able to reason about the material using simple representations. This leads the robot to demonstrate more accurate or efficient manipulation of that material.

We begin our exploration of interactive perception in Part I by examining liquids inside a container. In Chapter 2, analytic equations of liquids at steady state and sloshing liquids reveal that there is a direct connection between material properties of the liquid and forces and torques experienced by the container. From this observation, it is clear that an effective interactive perception framework for this system would be composed of a Force/Torque sensor mounted at the wrist of the robot, paired with intentional slow and fast rotations of the container. This sensor-interaction combination enables the robot to estimate the mass, volume, and viscosity of liquids inside of a container. Furthermore, using the estimate of volume, the robot is able to precisely pour a desired amount of water from the container.

Following this work, Part II studies the perception of stochastic dynamics in the form of granular materials and a single, bouncing ball. Granular material modeling is still an active area of research, so for Part II of this thesis, we look beyond analytic equations to model these materials. We first note that the material properties and stochastic dynamics experienced by grains and a bouncing ball are directly related to collision dynamics. Thus, Part II involves robotic interactions that intentionally induce these collisions. Given that the collisions reflect the material properties of the grains and bouncing ball, we explore two different approaches in Part II. In Chapter 3, we deduce that the steady state of an aggregate of grains reflects their dynamic history, and thus we choose a single, point-cloud of a steady-state granular formation as the observation for the numerous collisions that result in that formation. In Chapter 4, each ball collision is explicitly observed, so sound is deemed a sensible modality for making real-time observations of the bouncing ball.

Ultimately, these observations are used within a similar framework to automatically calibrate a simulation representation for the material or object. The method underlying this technology, a likelihood-free Bayesian inference approach called BayesSim, is shown to be effective at tuning simulators to match real-life distributions. Using this simulation representation of the granular materials and bouncing ball, Part II demonstrates how a robot could thus reason about the materials to pour grains into a certain pattern or to predict the behavior of a bouncing ball in order to deflect it in real-time.

Part III steers in a different direction in order to effectively perceive doughs by designing a series of variable-stiffness tactile sensors. Traditional tactile sensors are observed to be insufficient for perceiving deformable solid objects like dough. Tactile sensors composed of rigid components deform very soft objects, thus unable to measure geometric features or contact geometry. Meanwhile, compliant tactile sensors often lack the resolution to measure contact geometry. Thus, Part III presents three different iterations of soft, optical tactile sensors, which are compliant enough to measure the geometric features of soft objects without deforming them in the process. The developed sensors pivot from prior soft, optical tactile sensors in that they incorporate a variable-stiffness compliant membrane. Chapter 5 studies the sensing-actuation duality that is intrinsic to this variable-stiffness mechanism. Chapter 6 expands on this idea by modifying the design of the sensor in Chapter 5, making it easier to manufacture and integrate with an industry-standard robotic parallel-jaw gripper. Finally, Chapter 7 uses a modified version of the sensor from Chapter 6 to manipulate balls of dough into logs of varying lengths. Chapters 6 and 7 use the variable-stiffness tactile sensor to measure a proxy for stiffness of dough, which enables more effective manipulation of the dough. Furthermore, as the dynamic interaction between a soft membrane and dough is quite complex and difficult to model, Chapter 7 explores the use of a data-driven model to capture the dynamic interactions of the sensor and deformable solid object.

We hope this thesis will serve as inspiration for new ideas in robotic perception and manipulation of unstructured or deformable materials like liquids, grains, and doughs. In particular, we look forward to the following future directions:

Multi-Modal Sensor Fusion: This thesis applies various sensory modalities – some under-explored in literature, others employing new design principles – and demonstrates the many uses of these sensory frameworks for the perception of complex materials such as liquids and dough. To better equip robots with richer state-knowledge of these materials, we believe this technology would be most effective when used together as well as combined with existing visual systems. For instance, the task of pouring cereal into a bowl involves several sensory signals – tactile and force sensing instruct how the cereal box must be agitated to sufficiently shake out grains of cereal; visual sensing generates a feedback signal to avoid spilling cereal outside the bowl; sound signals shift as the cereal begins piling onto itself instead of hitting the ceramic surface of the bowl. The incorporation of multiple sensor modalities could lead to more robust state estimations and predictions, and, consequently, more robust manipulation capabilities. Furthermore, drawing from a diverse set of sensors enables robots to use slow but high-dimensional observations (e.g., video) for some subtasks and fast but low-dimensional observations (e.g., sound) for other subtasks. One

takeaway from this thesis is that there is no "perfect" sensor that is adequate for all tasks. Instead, a more holistic approach would be to employ the "best" sensor for a given interaction and purpose. Integrating many sensors allows a robot to experience the most informative signals it can observe during any task.

Exploring New Forms of Representation: In this thesis, we explore representations ranging from explicit analytic physics models to simulation representations to data-driven models. Future work may benefit from studying different methods for encoding the rich signals observed by the various types of perception frameworks presented in this thesis. For instance, while Chapter 4 demonstrates how sound is an effective sensory modality for real-time robotic applications, this work only grazes the surface of the rich information underlying vibrations and sound. We thus pose the questions: *What forms of representations can capture the most informative features of sound signals? What forms of representations are most robust to noise factors from the environment or sensors? How do these features change depending on the task? Do these representations enable explicit modeling of the material or do they require building a model through prior experience or data? Can these representations be encoded and interpreted without introducing significant latency to the real-time robotic system?* While these serve as example considerations to take when exploring the modality of sound, similar questions should be posed for other perception frameworks. As with the work in this thesis, exploring different representations may lead to new insights into the materials (e.g., Chapter 2) or sensors (e.g., Chapter 7) themselves.

Dynamic Manipulation of Unstructured Materials: Lastly, while this thesis primarily focuses on the *perception* of unstructured materials, we envision that the developed frameworks may lead to improved *manipulation* of these materials. In all the presented robotic demonstrations, many aspects of the robotic action space are fixed, thus greatly simplifying control. However, in order to match the fluidity of motion present in human-levels of dexterity, we believe investigating in more dynamic control strategies can lead to more efficient and effective manipulation capabilities. For instance, in Chapter 2, a simple rotation is applied to the container to pour a specific amount of liquid. However, the same volume of a liquid can be poured in different amounts of time depending on the motion of the robotic hand. *What dynamic control strategy could be used to optimize for time versus spillage?* Another example that would require a more sophisticated control strategy is sifting grains from a box. *What sort of sensory modality could be used as feedback to precisely sift out a specific amount of grains, and how might this dynamic motion be learned?* Finally, in Chapter 7, dough is rolled out into logs of varying lengths. While the reinforcement learning framework demonstrates improvements in efficiency when compared to a heuristic policy, the action space is limited to back-and-forth motions along the axis of the StRETcH sensor. However, human bakers exhibit highly dynamic motions when rolling and shaping dough, and these motions efficiently manipulate dough at rates that are magnitudes faster. We hope that, with new tools such as the StRETcH end-effector, future studies in dough manipulation can make use of a richer action space to interact more dynamically with these unstructured materials.

Bibliography

[1] Adam Allevato et al. “TuneNet: One-Shot Residual Tuning for System Identification and Sim-to-Real Robot Task Transfer”. In: *Conference on Robot Learning*. 2019, pp. 445–455.

[2] Alex Alspach et al. “Soft-bubble: A highly compliant dense geometry tactile sensor for robot manipulation”. In: *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE. 2019, pp. 597–604.

[3] L Anand and C Gu. “Granular materials: constitutive equations and strain localization”. In: *Journal of the Mechanics and Physics of Solids* 48.8 (2000), pp. 1701–1733.

[4] Martin Asenov et al. “Vid2Param: Modelling of Dynamics Parameters from Video”. In: *IEEE Robotics and Automation Letters* (2019).

[5] Karl Johan Åström and Peter Eykhoff. “System identification—a survey”. In: *Automatica* 7.2 (1971), pp. 123–162.

[6] Asli Atalay et al. “Batch fabrication of customizable silicone-textile composite capacitive strain sensors for human motion tracking”. In: *Advanced Materials Technologies* 2.9 (2017), p. 1700136.

[7] Ruzena Bajcsy. “Active perception”. In: *Proceedings of the IEEE* 76.8 (1988), pp. 966–1005.

[8] Devin J Balkcom and Matthew T Mason. “Robotic origami folding”. In: *The International Journal of Robotics Research* 27.5 (2008), pp. 613–627.

[9] Maria Bauza, Francois R Hogan, and Alberto Rodriguez. “A data-efficient approach to precise and controlled pushing”. In: *Conference on Robot Learning*. 2018.

[10] Maria Bauza and Alberto Rodriguez. “A probabilistic data-driven model for planar pushing”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 3008–3015.

[11] Stefan Begej. “Planar and finger-shaped optical tactile sensors for robotic applications”. In: *IEEE Journal on Robotics and Automation* 4.5 (1988), pp. 472–484.

[12] Matthew R Begley and Thomas J Mackin. “Spherical indentation of freestanding circular thin films in the membrane regime”. In: *Journal of the Mechanics and Physics of Solids* 52.9 (2004), pp. 2005–2023.

[13] L Benvenuti, C Kloss, and S Pirker. "Identification of DEM simulation parameters by Artificial Neural Networks and bulk experiments". In: *Powder Technology* 291 (2016), pp. 456–465.

[14] Alan D Bernstein. "Listening to the coefficient of restitution". In: *American Journal of Physics* 45.1 (1977), pp. 41–44.

[15] Narain Mulchand Bhatia and W Nachbar. "Finite indentation of an elastic membrane by a spherical indenter". In: *International Journal of Non-Linear Mechanics* 3.3 (1968), pp. 307–324.

[16] Jeannette Bohg et al. "Interactive perception: Leveraging action in perception and perception in action". In: *IEEE Transactions on Robotics* 33.6 (2017), pp. 1273–1291.

[17] Zdravko I Botev et al. "The cross-entropy method for optimization". In: *Handbook of Statistics*. Vol. 31. Elsevier, 2013, pp. 35–59.

[18] Christopher A Brooks and Karl Iagnemma. "Vibration-based terrain classification for planetary exploration rovers". In: *IEEE Transactions on Robotics* 21.6 (2005), pp. 1185–1191.

[19] Eric Brown et al. "Universal robotic gripper based on the jamming of granular material". In: *Proceedings of the National Academy of Sciences* 107.44 (2010), pp. 18809–18814.

[20] Gereon H Büscher et al. "Flexible and stretchable fabric-based tactile sensor". In: *Robotics and Autonomous Systems* 63 (2015), pp. 244–252.

[21] Claudia Carello, Krista L Anderson, and Andrew J Kunkler-Peck. "Perception of object length by sound". In: *Psychological Science* 9.3 (1998), pp. 211–214.

[22] Yevgen Chebotar et al. "Closing the sim-to-real loop: Adapting simulation randomization with real world experience". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8973–8979.

[23] Carolyn L Chen, Jeffrey O Snyder, and Peter J Ramadge. "Learning to identify container contents through tactile vibration signatures". In: *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pp. 43–48.

[24] Hongyang Cheng et al. "An iterative Bayesian filtering framework for fast and automated calibration of DEM models". In: *Computer Methods in Applied Mechanics and Engineering* 350 (2019), pp. 268–294.

[25] Andrea Cherubini et al. "Model-free vision-based shaping of deformable plastic materials". In: *IJRR* (2020), p. 0278364920907684.

[26] Sachin Chitta, Matthew Piccoli, and Jürgen Sturm. "Tactile object class and internal state recognition for mobile manipulation". In: *2010 IEEE International Conference on Robotics and Automation*, pp. 2342–2348.

[27] Craig Chorley et al. "Development of a tactile sensor based on biologically inspired edge encoding". In: *Advanced Robotics, 2009. ICAR 2009. International Conference on*. IEEE. 2009, pp. 1–6.

[28] Joshua Christie and Navinda Kottege. "Acoustics based terrain classification for legged robots". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 3596–3603.

[29] Matteo Cianchetti et al. "Soft robotics technologies to address shortcomings in today's minimally invasive surgery: the STIFF-FLOP approach". In: *Soft Robotics* 1.2 (2014), pp. 122–131.

[30] Pierluigi Cigliano et al. "Robotic ball catching with an eye-in-hand single-camera system". In: *IEEE Transactions on Control Systems Technology* 23.5 (2015), pp. 1657–1671.

[31] Samuel Clarke et al. "Learning Audio Feedback for Estimating Amount and Flow of Granular Material". In: *Conference on Robot Learning*. 2018, pp. 529–550.

[32] CJ Coetzee. "Calibration of the discrete element method". In: *Powder Technology* 310 (2017), pp. 104–142.

[33] Luke Cramphorn, Benjamin Ward-Cherrier, and Nathan F Lepora. "Tactile manipulation with biomimetic active touch". In: *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE. 2016, pp. 123–129.

[34] Ana-Maria Cretu, Pierre Payeur, and Emil M Petriu. "Soft object deformation monitoring and learning for model-based robotic hand manipulation". In: *IEEE Cybernetics* 42.3 (2011), pp. 740–753.

[35] Ana-Maria Cretu et al. "Deformable object segmentation and contour tracking in image sequences using unsupervised networks". In: *2010 Canadian Conference on Computer and Robot Vision*. IEEE. 2010, pp. 277–284.

[36] Peter A Cundall and Otto DL Strack. "A discrete numerical model for granular assemblies". In: *Geotechnique* 29.1 (1979), pp. 47–65.

[37] Michael Dawson-Haggerty. *Trimesh*. `https://github.com/mikedh/trimesh/blob/master/LICENSE.md`. Accessed: 2019-02-20.

[38] Karola Dierichs, Tobias Schwinn, and Achim Menges. "Robotic Pouring of Aggregate Structures". In: *Rob— Arch 2012*. Springer, 2013, pp. 196–205.

[39] Mohammad Hassan Djavareshkian et al. "Simulation of sloshing with the volume of fluid method". In: *Fluid Dynamics and Material Processing* 2.4 (2006), pp. 299–308.

[40] Chau Do, Tobias Schubert, and Wolfram Burgard. "A probabilistic approach to liquid level detection in cups using an RGB-D camera". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2075–2080.

[41] Huy Q Do, Alejandro M Aragón, and Dingena L Schott. "Automated discrete element method calibration using genetic and optimization algorithms". In: *EPJ Web of Conferences*. Vol. 140. EDP Sciences. 2017, p. 15011.

[42] Michal Karol Dobrzynski, Ramon Pericet-Camara, and Dario Floreano. "Contactless deflection sensor for soft robots". In: *Soft Robotics* 1.1 (2013), pp. 53–62.

[43] Franklin T Dodge et al. *The new" dynamic behavior of liquids in moving containers"*. Southwest Research Inst. San Antonio, TX, 2000.

[44] Alin Drimus et al. "Classification of rigid and deformable objects using a novel tactile sensor". In: *2011 15th International Conference on Advanced Robotics (ICAR)*. IEEE. 2011, pp. 427–434.

[45] Sachith Dunatunga and Ken Kamrin. "Continuum modelling and simulation of granular flows through their many phases". In: *Journal of Fluid Mechanics* 779 (2015), pp. 483–513.

[46] Frederik Ebert et al. "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control". In: *arXiv preprint arXiv:1812.00568* (2018).

[47] Christof Elbrechter et al. "Discriminating liquids using a robotic kitchen assistant". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 703–708.

[48] Ronald S Fearing. "Tactile sensing mechanisms". In: *The International Journal of Robotics Research* 9.3 (1990), pp. 3–23.

[49] John T Feddema et al. "Control for slosh-free motion of an open container". In: *IEEE Control Systems Magazine* 17.1 (1997), pp. 29–36.

[50] Nicola J Ferrier and Roger W Brockett. "Reconstructing the shape of a deformable membrane from image data". In: *The International Journal of Robotics Research* 19.9 (2000), pp. 795–816.

[51] Nadia Figueroa, Ana Lucia Pais Ureche, and Aude Billard. "Learning complex sequential tasks from demonstration: A pizza dough rolling case study". In: *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Ieee. 2016, pp. 611–612.

[52] Kevin C Galloway et al. "Soft robotic grippers for biological sampling on deep reefs". In: *Soft robotics* 3.1 (2016), pp. 23–33.

[53] Aditya Ganapathi et al. "Learning to smooth and fold real fabric using dense object descriptors trained on synthetic color images". In: *arXiv preprint arXiv:2003.12698* (2020).

[54] Dhiraj Gandhi, Abhinav Gupta, and Lerrel Pinto. "Swoosh! Rattle! Thump!-Actions that Sound". In: *RSS*. 2020.

[55] Piiren Giiler et al. "Estimating the deformability of elastic materials using optical flow and position-based dynamics". In: *2015 IEEE-RAS 15th Humanoids*. IEEE. 2015, pp. 965–971.

[56] Kenneth Y Goldberg and Ruzena Bajcsy. "Active touch and robot perception". In: *Cognition and Brain Theory* 7.2 (1984), pp. 199–214.

[57] Sebastian Gomez-Gonzalez et al. "Reliable Real-Time Ball Tracking for Robot Table Tennis". In: *Robotics* 8.4 (2019), p. 90.

[58] *Granular Material by Design*. `https://jfi.uchicago.edu/~jaeger/group/Granular_Matter_by_Design/Granular_Matter_by_Design.html`. Accessed:2019-09-15.

[59] Massimo Grassi. “Do we hear size or sound? Balls dropped on plates”. In: *Perception & Psychophysics* 67.2 (2005), pp. 274–284.

[60] Püren Güler et al. “What’s in the container? Classifying object contents from vision and touch”. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3961–3968.

[61] Frank L Hammond et al. “Soft tactile sensor arrays for micromanipulation”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 25–32.

[62] Haifeng Han, Gavin Paul, and Takamitsu Matsubara. “Model-based reinforcement learning approach for deformable linear object manipulation”. In: *13th IEEE CASE*. 2017, pp. 750–755.

[63] Michael Heckel et al. “Can we obtain the coefficient of restitution from the sound of a bouncing ball?” In: *Physical Review E* 93.3 (2016), p. 032901.

[64] Robert Alan Hess et al. *Liquid gauging using sensor fusion and data fusion*. US Patent 6,157,894. 2000.

[65] Mitsuru Higashimori, Kayo Yoshimoto, and Makoto Kaneko. “Active shaping of an unknown rheological object based on deformation decomposition into elasticity and plasticity”. In: *2010 IEEE ICRA*. IEEE. 2010, pp. 5120–5126.

[66] Andrew D Hinitt, Jonathan Rossiter, and Andrew T Conn. “WormTIP: An Invertebrate Inspired Active Tactile Imaging Pneumostat”. In: *Conference on Biomimetic and Biohybrid Systems*. Springer. 2015, pp. 38–49.

[67] EGM Holweg et al. “Slip detection by tactile sensors: Algorithms and experimental results”. In: *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*. Vol. 4. IEEE. 1996, pp. 3234–3239.

[68] Robert D Howe and Mark R Cutkosky. “Sensing skin acceleration for slip and texture perception”. In: *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*. IEEE. 1989, pp. 145–150.

[69] Dimitris Hristu, Nicola Ferrier, and Roger W Brockett. “The performance of a deformable-membrane tactile sensor: basic results on geometrically-defined tasks”. In: *Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on*. Vol. 1. IEEE. 2000, pp. 508–513.

[70] Yuanming Hu et al. “A moving least squares material point method with displacement discontinuity and two-way rigid body coupling”. In: *ACM Transactions on Graphics (TOG)* 37.4 (2018), p. 150.

[71] Isabella Huang and Ruzena Bajcsy. “High Resolution Soft Tactile Interface for Physical Human-Robot Interaction”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 1705–1711.

[72] Isabella Huang and Ruzena Bajcsy. "Robot Learning from Demonstration with Tactile Signals for Geometry-Dependent Tasks". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020.

[73] Isabella Huang, Jingjun Liu, and Ruzena Bajcsy. "A Depth Camera-Based Soft Fingertip Device for Contact Region Estimation and Perception-Action Coupling". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8443–8449.

[74] Tae Myung Huh et al. "A Multi-Chamber Smart Suction Cup for Adaptive Gripping and Haptic Exploration". In: *arXiv preprint arXiv:2105.02345* (2021).

[75] Raouf A Ibrahim. *Liquid sloshing dynamics: theory and applications*. Cambridge University Press, 2005.

[76] Filip Ilievski et al. "Soft robotics for chemists". In: *Angewandte Chemie* 123.8 (2011), pp. 1930–1935.

[77] Heinrich M Jaeger and Sidney R Nagel. "Physics of the granular state". In: *Science* 255.5051 (1992), pp. 1523–1531.

[78] Rishabh Jangir, Guillem Alenyà, and Carme Torras. "Dynamic cloth manipulation with deep reinforcement learning". In: *2020 IEEE ICRA*. IEEE. 2020, pp. 4630–4636.

[79] Kenneth Langstreth Johnson. *Contact mechanics*. Cambridge university press, 1987.

[80] Lynette A Jones and Susan J Lederman. *Human hand function*. Oxford university press, 2006.

[81] Pierre Jop, Yoël Forterre, and Olivier Pouliquen. "A constitutive law for dense granular flows". In: *Nature* 441.7094 (2006), p. 727.

[82] Feng Ju et al. "A variable-impedance piezoelectric tactile sensor with tunable sensing performance for tissue hardness sensing in robotic tumor palpation". In: *Smart Materials and Structures* 27.11 (2018), p. 115039.

[83] Kwangmok Jung et al. "Artificial annelid robot driven by soft actuators". In: *Bioinspiration & Biomimetics* 2.2 (2007), S42.

[84] Mohsen Kaboli, Kunpeng Yao, and Gordon Cheng. "Tactile-based manipulation of deformable objects with dynamic center of mass". In: *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE. 2016, pp. 752–757.

[85] Ken Kamrin. "Non-locality in Granular Flow: Phenomenology and modeling approaches". In: *Frontiers in Physics* 7 (2019), p. 116.

[86] Ken Kamrin. "Quantitative Rheological Model for Granular Materials: The Importance of Particle Size". In: *Handbook of Materials Modeling: Applications: Current and Emerging Materials* (2018), pp. 1–24.

[87] Ken Kamrin and Georg Koval. "Effect of particle surface friction on nonlocal constitutive behavior of flowing granular media". In: *Computational Particle Mechanics* 1.2 (2014), pp. 169–176.

[88] Dov Katz and Oliver Brock. "Manipulating articulated objects with interactive perception". In: *2008 IEEE International Conference on Robotics and Automation*, pp. 272–277.

[89] Dov Katz et al. "Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects". In: *2013 IEEE International Conference on Robotics and Automation*, pp. 5003–5010.

[90] Monroe Kennedy et al. "Precise dispensing of liquids using visual feedback". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1260–1266.

[91] Roberta L Klatzky, Dinesh K Pai, and Eric P Krotkov. "Perception of material from contact sounds". In: *Presence: Teleoperators & Virtual Environments* 9.4 (2000), pp. 399–410.

[92] Margaret A Knuth et al. "Discrete element modeling of a Mars Exploration Rover wheel in granular material". In: *Journal of Terramechanics* 49.1 (2012), pp. 27–36.

[93] Oliver Kroemer, Christoph H Lampert, and Jan Peters. "Learning dynamic tactile sensing with robust vision-based training". In: *IEEE Transactions on Robotics* 27.3 (2011), pp. 545–557.

[94] Eric Krotkov. "Robotic perception of material". In: *IJCAI*. 1995, pp. 88–95.

[95] Eric Krotkov, Roberta Klatzky, and Nina Zumel. "Robotic perception of material: Experiments with shape-invariant acoustic measures of material type". In: *Experimental Robotics IV*. Springer, 1997, pp. 204–211.

[96] James Kubricht et al. "Probabilistic Simulation Predicts Human Performance on Viscous Fluid-Pouring Problem." In: *CogSci*. 2016.

[97] Johannes W Kuehl and Jason W Capps. *System and method for measuring liquid mass quantity*. US Patent 5,138,559. 1992.

[98] Charles D Kuglin. "The phase correlation image alignment methed". In: *Proc. Int. Conference Cybernetics Society*. 1975, pp. 163–165.

[99] T Kuhn. *Electrondust*. 2018. URL: `https://electrondust.com/2018/07/22/stepper-juggler/`.

[100] Lars Kunze and Michael Beetz. "Envisioning the qualitative effects of robot manipulation actions using simulation-based projections". In: *Artificial Intelligence* 247 (2017), pp. 352–380.

[101] Naveen Kuppuswamy et al. "Soft-bubble grippers for robust and perceptive manipulation". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 9917–9924.

[102] Nathan O Lambert et al. "Low-level control of a quadrotor with deep model-based reinforcement learning". In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 4224–4230.

[103] Sebastian Lang et al. “Providing the basis for human-robot-interaction: A multi-modal attention system for a mobile robot”. In: *Proceedings of the 5th international conference on Multimodal interfaces*. 2003, pp. 28–35.

[104] William E Langlois and Michel O Deville. *Slow viscous flow*. Vol. 173436. Springer, 1964.

[105] Christina Larson et al. “Highly stretchable electroluminescent skin for optical signaling and tactile sensing”. In: *Science* 351.6277 (2016), pp. 1071–1074.

[106] Cecilia Laschi et al. “Soft robot arm inspired by the octopus”. In: *Advanced Robotics* 26.7 (2012), pp. 709–727.

[107] Susan J Lederman and Roberta L Klatzky. “Hand movements: A window into haptic object recognition”. In: *Cognitive psychology* 19.3 (1987), pp. 342–368.

[108] Alex X Lee et al. “Learning force-based manipulation of deformable objects from multiple demonstrations”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 177–184.

[109] Michelle A Lee et al. “Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8943–8950.

[110] Wang Wei Lee et al. “A neuro-inspired artificial peripheral nervous system for scalable electronic skins”. In: *Science Robotics* 4.32 (2019), eaax2198.

[111] Sergey Levine and Pieter Abbeel. “Learning Neural Network Policies with Guided Policy Search under Unknown Dynamics.” In: *NIPS*. Vol. 27. Citeseer. 2014, pp. 1071–1079.

[112] Chen Li, Tingnan Zhang, and Daniel I Goldman. “A terradynamics of legged locomotion on granular media”. In: *Science* 339.6126 (2013), pp. 1408–1412.

[113] Hailing Li et al. “Ping-pong robotics with high-speed vision system”. In: *2012 12th International Conference on Control Automation Robotics & Vision (ICARCV)*. IEEE. 2012, pp. 106–111.

[114] Rui Li et al. “Localization and manipulation of small parts using gelsight tactile sensing”. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE. 2014, pp. 3988–3993.

[115] Hongzhuo Liang et al. “Making Sense of Audio Vibration for Liquid Height Estimation in Robotic Pouring”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 5333–5339.

[116] Jacky Liang et al. “GPU-accelerated robotic simulation for distributed reinforcement learning”. In: *Conference on Robot Learning*. PMLR. 2018, pp. 270–282.

[117] Xingyu Lin et al. “SoftGym: Benchmarking Deep Reinforcement Learning for Deformable Object Manipulation”. In: *Conference on Robot Learning (CORL)*. 2020.

[118] Andrea J Liu and Sidney R Nagel. “Granular and jammed materials”. In: *Soft Matter* 6.13 (2010), pp. 2869–2870.

[119] Junjie Liu et al. “Puncture mechanics of soft elastomeric membrane with large deformation by rigid cylindrical indenter”. In: *Journal of the Mechanics and Physics of Solids* 112 (2018), pp. 458–471.

[120] Nanshu Lu and Kim Dae-Hyeong. “Flexible and stretchable electronics paving the way for soft robotics”. In: *Soft Robotics* 1.1 (2013), pp. 53–62.

[121] Shan Luo et al. “Robotic tactile perception of object properties: A review”. In: *Mechatronics* 48 (2017), pp. 54–67.

[122] Xubo Lyu et al. “MBB: Model-Based Baseline for Efficient Reinforcement Learning”. In: *arXiv preprint arXiv:2011.02073* (2020).

[123] Daolin Ma and Alberto Rodriguez. “Friction variability in planar pushing data: Anisotropic friction and data-collection bias”. In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3232–3239.

[124] Miles Macklin and Matthias Müller. “Position based fluids”. In: *ACM Transactions on Graphics (TOG)* 32.4 (2013), p. 104.

[125] Miles Macklin et al. “Non-smooth Newton methods for deformable multi-body dynamics”. In: *ACM Transactions on Graphics (TOG)* 38.5 (2019), pp. 1–20.

[126] Miles Macklin et al. “Unified particle physics for real-time applications”. In: *ACM Transactions on Graphics (TOG)* 33.4 (2014), p. 153.

[127] Ryan D Maladen et al. “Undulatory swimming in sand: experimental and simulation studies of a robotic sandfish”. In: *The International Journal of Robotics Research* 30.7 (2011), pp. 793–805.

[128] Roberto Martın-Martın and Oliver Brock. “Building kinematic and dynamic models of articulated objects with multi-modal interactive perception”. In: *AAAI Symposium on Interactive Multi-Sensory Object Perception for Embodied Agents, AAAI, Ed*. 2017.

[129] Jan Matas, Stephen James, and Andrew J Davison. “Sim-to-Real Reinforcement Learning for Deformable Object Manipulation”. In: *CORL*. 2018, pp. 734–743.

[130] Carolyn Matl and Ruzena Bajcsy. “Deformable Elasto-Plastic Object Shaping using an Elastic Hand and Model-Based Reinforcement Learning”. In: *2021 IEEE/RSJ IROS*. IEEE. 2021.

[131] Carolyn Matl, Josephine Koe, and Ruzena Bajcsy. “StRETcH: A Soft to Resistive Tactile Hand”. In: *2021 IEEE ICRA*. IEEE. 2021.

[132] Carolyn Matl, Robert Matthew, and Ruzena Bajcsy. “Haptic perception of liquids enclosed in containers”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 7142–7149.

[133] Carolyn Matl et al. “Inferring the Material Properties of Granular Media for Robotic Tasks”. In: *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2020.

[134] Carolyn Matl et al. “STReSSD: Sim-To-Real from Sound for Stochastic Dynamics”. In: *CoRL*. 2020, pp. 529–550.

[135] Hammad Mazhar et al. “Studying the effect of powder geometry on the selective laser sintering process”. In: *Society of Plastics Engineers (SPE) ANTEC* (2014).

[136] Benjamin W McInroe et al. “Towards a soft fingertip with integrated sensing and actuation”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 6437–6444.

[137] Claudio Melchiorri. “Slip detection and control using tactile and force sensors”. In: *IEEE/ASME transactions on mechatronics* 5.3 (2000), pp. 235–243.

[138] GN Mikishev and N Ya Dorozhkin. “Experimental study of free oscillations of liquid in vessels”. In: *Izv. AN SSSR, OTN, Mekhanika i mashinostroenie* 4 (1961).

[139] Stephen Miller et al. “A geometric approach to robotic laundry folding”. In: *The International Journal of Robotics Research* 31.2 (2012), pp. 249–267.

[140] Brian Mirtich. “Fast and accurate computation of polyhedral mass properties”. In: *Journal of Graphics Tools* 1.2 (1996), pp. 31–50.

[141] Marina Montaine et al. “Coefficient of restitution as a fluctuating quantity”. In: *Physical Review E* 84.4 (2011), p. 041306.

[142] Bobak Mosadegh et al. “Pneumatic networks for soft robotics that actuate rapidly”. In: *Advanced Functional Materials* 24.15 (2014), pp. 2163–2170.

[143] Gabriel Moy, C Wagner, and Ronald S Fearing. “A compliant tactile display for teletaction”. In: *Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on*. Vol. 4. IEEE. 2000, pp. 3409–3415.

[144] Matthias Müller, David Charypar, and Markus H Gross. “Particle-based fluid simulation for interactive applications”. In: *Symposium on Computer animation*, pp. 154–159.

[145] Matthias Müller et al. “Position based dynamics”. In: *Journal of Visual Communication and Image Representation* 18.2 (2007), pp. 109–118.

[146] John C Murray, Harry R Erwin, and Stefan Wermter. “Robotic sound-source localisation architecture using cross-correlation and recurrent neural networks”. In: *Neural Networks* 22.2 (2009), pp. 173–189.

[147] Félix Nadon, Angel J Valencia, and Pierre Payeur. “Multi-modal sensing and robotic manipulation of non-rigid objects: A survey”. In: *Robotics* 7.4 (2018), p. 74.

[148] Anusha Nagabandi et al. “Deep dynamics models for learning dexterous manipulation”. In: *CORL*. PMLR. 2020, pp. 1101–1112.

[149] Anusha Nagabandi et al. “Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning”. In: *2018 IEEE ICRA*. IEEE. 2018, pp. 7559–7566.

[150] Yashraj S Narang et al. “Interpreting and Predicting Tactile Signals via a Physics-Based and Data-Driven Framework”. In: *2020 Robotics: Science and Systems (RSS)*. 2020.

[151] Erickson R Nascimento et al. "BRAND: A robust appearance and depth descriptor for RGB-D images". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 1720–1726.

[152] David Navarro-Alarcon and Yun-Hui Liu. "Fourier-based shape servoing: a new feedback method to actively deform soft objects into desired 2-D image contours". In: *IEEE Transactions on Robotics* 34.1 (2017), pp. 272–279.

[153] David Navarro-Alarcon et al. "Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model". In: *IEEE Transactions on Robotics* 32.2 (2016), pp. 429–441.

[154] Dan Negrut et al. "Investigating through simulation the mobility of light tracked vehicles operating on discrete granular terrain". In: *SAE International Journal of Passenger Cars-Mechanical Systems* 6.2013-01-1191 (2013), pp. 369–381.

[155] Sia Nemat-Nasser and Juhua Zhang. "Constitutive relations for cohesionless frictional granular materials". In: *International Journal of Plasticity* 18.4 (2002), pp. 531–547.

[156] Yoshiyuki Noda et al. "Flow rate control based on differential flatness in automatic pouring robot". In: *2011 IEEE International Conference on Control Applications (CCA)*. IEEE, pp. 1468–1475.

[157] *NVIDIA Isaac: Virtual Simulator for Robots*. `https://www.nvidia.com/en-us/deep-learning-ai/industries/robotics/`. Accessed: 2019-09-11.

[158] Sadao Omata, Yoshinobu Murayama, and Christos E Constantinou. "Real time robotic tactile sensor system for the determination of the physical properties of biomaterials". In: *Sensors and Actuators A: Physical* 112.2-3 (2004), pp. 278–285.

[159] Djenane C Pamplona, Hans I Weber, and Guilherme R Sampaio. "Analytical, numerical and experimental analysis of continuous indentation of a flat hyperelastic circular membrane by a rigid cylindrical indenter". In: *International Journal of Mechanical Sciences* 87 (2014), pp. 18–25.

[160] Zherong Pan and Dinesh Manocha. "Motion planning for fluid manipulation using simplified dynamics". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4224–4231.

[161] Zherong Pan, Chonhyon Park, and Dinesh Manocha. "Robot motion planning for pouring liquids". In: *Twenty-Sixth International Conference on Automated Planning and Scheduling*. 2016.

[162] George Papamakarios and Iain Murray. "Fast ε-free inference of simulation models with bayesian conditional density estimation". In: *Advances in Neural Information Processing Systems*. 2016, pp. 1028–1036.

[163] Xue Bin Peng et al. "Sim-to-real transfer of robotic control with dynamics randomization". In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 1–8.

[164] Antoine Petit et al. "Tracking elastic deformable objects with an RGB-D sensor for a pizza chef robot". In: *Robotics and Autonomous Systems* 88 (2017), pp. 187–201.

[165] Alexander Podlozhnyuk, Stefan Pirker, and Christoph Kloss. "Efficient implementation of superquadric particles in Discrete Element Method within an open-source framework". In: *Computational Particle Mechanics* 4.1 (2017), pp. 101–118.

[166] Panagiotis Polygerinos et al. "Soft robotic glove for combined assistance and at-home rehabilitation". In: *Robotics and Autonomous Systems* 73 (2015), pp. 135–143.

[167] Xiong Pu et al. "Ultrastretchable, transparent triboelectric nanogenerator as electronic skin for biomechanical energy harvesting and tactile sensing". In: *Science advances* 3.5 (2017), e1700015.

[168] Ali Rahimi and Benjamin Recht. "Random features for large-scale kernel machines". In: *Advances in Neural Information Processing Systems*. 2008, pp. 1177–1184.

[169] Fabio Ramos, Rafael Carvalhaes Possas, and Dieter Fox. "BayesSim: Adaptive domain randomization via probabilistic inference for robotics simulators". In: *RSS*. 2019.

[170] Mahmut Reyhanoglu and Jaime Rubio Hervas. "Point-to-point liquid container transfer via a PPR robot with sloshing suppression". In: *2012 American Control Conference (ACC)*. IEEE, pp. 5490–5494.

[171] Patrick Richard et al. "Slow relaxation and compaction of granular systems". In: *Nature Materials* 4.2 (2005), p. 121.

[172] Carl G Ringwall and Allen W Case Jr. *Tactile Sensor*. US Patent 4,306,148. 1981.

[173] Joseph M Romano et al. "Human-inspired robotic grasp control with tactile sensing". In: *IEEE Transactions on Robotics* 27.6 (2011), pp. 1067–1079.

[174] Leonel Rozo, Pablo Jiménez, and Carme Torras. "Force-based robot learning of pouring skills using parametric hidden markov models". In: *9th International Workshop on Robot Motion and Control*. IEEE. 2013, pp. 227–232.

[175] Daniela Rus and Michael T Tolley. "Design, fabrication and control of soft robots". In: *Nature* 521.7553 (2015), p. 467.

[176] Hannes Saal, Jo-Anne Ting, and Sethu Vijayakumar. "Active sequential learning with tactile feedback". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 2010, pp. 677–684.

[177] Jose Sanchez et al. "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey". In: *The International Journal of Robotics Research* 37.7 (2018), pp. 688–716.

[178] Connor Schenck and Dieter Fox. "Perceiving and reasoning about liquids using fully convolutional networks". In: *The International Journal of Robotics Research* 37.4-5 (2018), pp. 452–471.

[179] Connor Schenck and Dieter Fox. "Reasoning about liquids via closed-loop simulation". In: *2017 Robotics: Science & Systems (RSS)*. 2017.

[180] Connor Schenck and Dieter Fox. "Visual closed-loop control for pouring liquids". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2629–2636.

[181] Connor Schenck et al. "Learning robotic manipulation of granular media". In: *Conference on Robot Learning*. PMLR. 2017, pp. 239–248.

[182] ON Scott et al. "Indentation of freestanding circular elastomer films using spherical indenters". In: *Acta Materialia* 52.16 (2004), pp. 4877–4885.

[183] Dylan S Shah et al. "Morphing robots using robotic skins that sculpt clay". In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 2204–2211.

[184] Mitsuhiro Shikida et al. "Active tactile sensor for detecting contact force and hardness of an object". In: *Sensors and Actuators A: Physical* 103.1-2 (2003), pp. 213–218.

[185] Jivko Sinapov, Connor Schenck, and Alexander Stoytchev. "Learning relational object categories using behavioral exploration and multimodal perception". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 5691–5698.

[186] Pramath R Sinha and Ruzena Bajcsy. "Active exploration of surfaces for legged locomotion of robots". In: *Technical Reports (CIS)* (1990), p. 355.

[187] Jae S Son, Eduardo A Monteverde, and Robert D Howe. "A tactile sensor for localizing transient events in manipulation". In: *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*. IEEE. 1994, pp. 471–476.

[188] Mariacarla Staffa et al. "Segmentation performance in tracking deformable objects via WNNs". In: *IEEE ICRA*. 2015, pp. 2462–2467.

[189] Ivan Stensgaard and Erik Lægsgaard. "Listening to the coefficient of restitution-revisited". In: *American Journal of Physics* 69.3 (2001), pp. 301–305.

[190] Auston Sterling et al. "ISNN: Impact sound neural network for audio-visual object classification". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 555–572.

[191] Agostino Stilli et al. "Variable stiffness link (vsl): Toward inherently safe robotic manipulators". In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE. 2017, pp. 4971–4976.

[192] Jan Stria et al. "Garment perception and its folding using a dual-arm robot". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2014, pp. 61–67.

[193] Zhe Su et al. "Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor". In: *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*. IEEE. 2015, pp. 297–303.

[194] Chansu Suh et al. “Soft pneumatic actuator skin with embedded sensors”. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Ieee. 2014, pp. 2783–2788.

[195] Priya Sundaresan et al. “Learning rope manipulation policies using dense object descriptors trained on synthetic depth data”. In: *2020 IEEE ICRA*. IEEE. 2020, pp. 9411–9418.

[196] Daisuke Takagi and Herbert E Huppert. “Pouring viscous fluid out of a tipped container in minimal time”. In: *Physical Review E* 84.3 (2011), p. 035303.

[197] M Taylor and M Shirani. “Simulation of wrinkling in incompressible anisotropic thin sheets with wavy fibers”. In: *International Journal of Non-Linear Mechanics* (2020), p. 103610.

[198] M Taylor et al. “Finite elastic wrinkling deformations of incompressible fiber-reinforced plates”. In: *International Journal of Engineering Science* 144 (2019), p. 103138.

[199] Brijen Thananjeyan et al. “Multilateral surgical pattern cutting in 2d orthotropic gauze with deep reinforcement learning policies for tensioning”. In: *IEEE ICRA*. IEEE. 2017, pp. 2371–2378.

[200] Elliot Thompson-Bean, Oliver Steiner, and Andrew McDaid. “A soft robotic exoskeleton utilizing granular jamming”. In: *2015 IEEE International Conference On Advanced Intelligent Mechatronics (AIM)*. IEEE. 2015, pp. 165–170.

[201] Michael T Tolley et al. “A resilient, untethered soft robot”. In: *Soft Robotics* 1.3 (2014), pp. 213–223.

[202] Laura K Treers, Cyndia Cao, and Hannah S Stuart. “Granular Resistive Force Theory Implementation for Three-Dimensional Trajectories”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1887–1894.

[203] Dominic Vella et al. “The indentation of pressurized elastic shells: from polymeric capsules to yeast cells”. In: *Journal of the Royal Society Interface* (2011), rsif20110352.

[204] Alex Villanueva, Colin Smith, and Shashank Priya. “A biomimetic robotic jellyfish (Robojelly) actuated by shape memory alloy composite actuators”. In: *Bioinspiration & biomimetics* 6.3 (2011), p. 036004.

[205] Liyu Wang and Fumiya Iida. “Deformation in soft-matter robotics: A categorization and quantitative characterization”. In: *IEEE Robotics & Automation Magazine* 22.3 (2015), pp. 125–139.

[206] Benjamin Ward-Cherrier et al. “The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies”. In: *Soft Robotics* 5.2 (2018), pp. 216–227.

[207] Michael Wehner et al. “An integrated design and fabrication strategy for entirely soft, autonomous robots”. In: *Nature* 536.7617 (2016), pp. 451–455.

[208] Grady Williams et al. “Information theoretic MPC for model-based reinforcement learning”. In: *2017 IEEE ICRA*. IEEE. 2017, pp. 1714–1721.

[209] Bryan Willimon, Stan Birchfield, and Ian Walker. “Interactive perception of rigid and non-rigid objects”. In: *International Journal of Advanced Robotic Systems* 9.6 (2012), p. 227.

[210] Akihiko Yamaguchi and Christopher G Atkeson. "Combining finger vision and optical tactile sensing: Reducing and handling errors while cutting vegetables". In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE. 2016, pp. 1045–1051.

[211] Akihiko Yamaguchi and Christopher G Atkeson. "Implementing tactile behaviors using FingerVision". In: *Humanoid Robotics (Humanoids), 2017 IEEE-RAS 17th International Conference on*. IEEE. 2017, pp. 241–248.

[212] Akihiko Yamaguchi and Christopher G Atkeson. "Stereo vision of liquid and particle flow for robot pouring". In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, pp. 1173–1180.

[213] Hideyasu Yamashita et al. "Function-selectable tactile sensing system with morphological change". In: *System Integration (SII), 2016 IEEE/SICE International Symposium on*. IEEE. 2016, pp. 415–420.

[214] Wilson Yan et al. "Learning Predictive Representations for Deformable Objects Using Contrastive Estimation". In: *CORL*. 2020.

[215] Wei Yang, ShuHong Liu, and Hong Lin. "Viscous liquid sloshing damping in cylindrical container using a volume of fluid method". In: *Science in China Series E: Technological Sciences* 52.6 (2009), pp. 1484–1492.

[216] WH Yang and KH Hsu. "Indentation of a circular membrane". In: *ASME Journal of Applied Mechanics* (1971).

[217] Hang Yin, Anastasia Varava, and Danica Kragic. "Modeling, learning, perception, and control methods for deformable object manipulation". In: *Science Robotics* 6.54 (2021).

[218] Tomoaki Yoshikai et al. "Development of soft stretchable knit sensor for humanoids' whole-body tactile sensibility". In: *2009 9th IEEE-RAS International Conference on Humanoid Robots*. IEEE. 2009, pp. 624–631.

[219] Kayo Yoshimoto et al. "Active outline shaping of a rheological object based on plastic deformation distribution". In: *2011 IEEE/RSJ IROS*. IEEE. 2011, pp. 1386–1391.

[220] Hanna Yousef, Mehdi Boukallel, and Kaspar Althoefer. "Tactile sensing for dexterous in-hand manipulation in robotics—A review". In: *Sensors and Actuators A: physical* 167.2 (2011), pp. 171–187.

[221] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. "Gelsight: High-resolution robot tactile sensors for estimating geometry and force". In: *Sensors* 17.12 (2017), p. 2762.

[222] Wenzhen Yuan et al. "Active clothing material perception using tactile sensing and deep learning". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1–8.

[223] Wenzhen Yuan et al. "Measurement of shear and slip with a GelSight tactile sensor". In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 304–311.

[224] Wenzhen Yuan et al. “Shape-independent hardness estimation using deep learning and a gelsight tactile sensor”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 951–958.

[225] Hanafiah Yussof et al. “Determination of object stiffness control parameters in robot manipulation using a prototype optical three-axis tactile sensor”. In: *SENSORS, 2008 IEEE*. IEEE. 2008, pp. 992–995.

[226] Cha Zhang and Tsuhan Chen. “Efficient feature extraction for 2D/3D objects in mesh representation”. In: *Proceedings 2001 International Conference on Image Processing*. Vol. 3. IEEE, pp. 935–938.

[227] Qiong Zhang and Ken Kamrin. “Microscopic description of the granular fluidity field in nonlocal flow modeling”. In: *Physical Review Letters* 118.5 (2017), p. 058001.

[228] Tingnan Zhang et al. “Ground fluidization promotes rapid running of a lightweight robot”. In: *The International Journal of Robotics Research* 32.7 (2013), pp. 859–869.

[229] Zhoutong Zhang et al. “Shape and material from sound”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 1278–1288.

[230] Jihong Zhu et al. “Challenges and Outlook in Robotic Manipulation of Deformable Objects”. In: *arXiv preprint arXiv:2105.01767* (2021).

[231] Jihong Zhu et al. “Dual-arm robotic manipulation of flexible cables”. In: *2018 IEEE/RSJ IROS*. IEEE. 2018, pp. 479–484.

[232] Yifan Zhu, Laith Abdulmajeid, and Kris Hauser. “A Data-driven Approach for Fast Simulation of Robot Locomotion on Granular Media”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 7653–7659.

Appendix A

Derivations for Chapter 2

A.1 Liquid Mass

Given the force oriented in the positive y direction, f_y, of the static world frame, which aligns with the negative of the gravitational force vector, then f_y is due to the collective mass of the end-effector attachment, i.e., the mass of the container m_c, the mass of the gripper m_g, and the mass of the contained liquid, m_ℓ. We want to find m_ℓ, and all other quantities are known, so:

$$\begin{aligned} f_y &= g(m_c + m_g + m_\ell) \\ {}^{f_y}/_{g} - m_g - m_c &= m_\ell \end{aligned}$$

A.2 Liquid Volume

Here, we derive the piecewise continuous equations that define the analytic model of liquid volume in a circular cylindrical container. First, we rewrite the torque τ_z about the wrist of the robot in terms of the centers of mass of the attached system (see Figure A.1):

$$\begin{aligned} \tau_z &= f_y \bar{x}_M \\ &= g(m_g + m_c + m_\ell)\frac{m_g\bar{x}_g + m_c\bar{x}_c + m_\ell\bar{x}_\ell}{m_g + m_c + m_\ell} \\ &= g(m_g\bar{x}_g + m_c\bar{x}_c + m_\ell\bar{x}_\ell) \end{aligned}$$

As depicted in Figure A.1, the world frame is anchored at the center of the gripper. Thus, $\bar{x}_g = 0$. Since we assume that the container has a shell with a uniform density, then relative to the world frame, $\bar{x}_c = ({}^{L}/_{2} - L_g)\sin\theta$. Note that $\bar{x}_\ell$ will be a function of the rotation angle θ, the volume V, and the container geometry. Thus, we get an expression of τ_z as a function of θ and V:

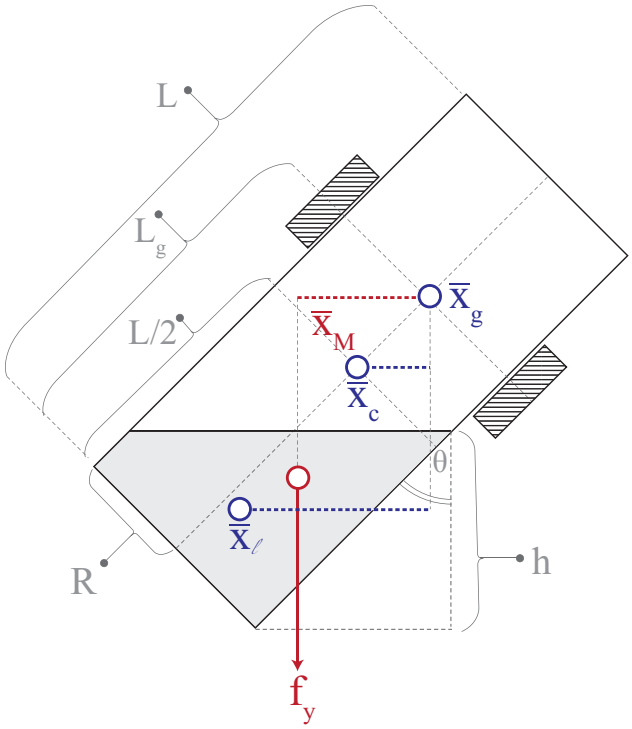


Figure A.1: A cross-sectional diagram defining variables used to derive liquid volume calculations. Example locations of center of masses of the container, gripper, and liquid are illustrated.

$$\begin{aligned}\tau_z(\theta) &= g(m_g * 0 + m_c * ((L/2 - L_g)\sin\theta) + m_\ell * \bar{x}_\ell(\theta, V)) \\ &= g(m_c((L/2 - L_g)\sin\theta) + m_\ell \bar{x}_\ell(\theta, V))\end{aligned}$$

Finding V becomes an optimization problem, where we want to minimize the residuals between the measured torques at different rotation angles $\tau_z(\theta)$ and our model-based estimated values:

$$V = \arg\min_V \left\| \tau_z(\theta) - g(m_c(L/2 - L_g)\sin\theta + m_\ell \bar{x}_\ell(\theta, V)) \right\|_2^2$$

Below, we show full derivations of how we define the piecewise nonlinear function $\bar{x}_\ell(\theta, V)$. All calculations of $\bar{x}_\ell^{B*}$ and $\bar{y}_\ell^{B*}$ are made with respect to a shifted body frame $B*$, where the origin is centered at the bottom face of the cylindrical container (see Figure A.2). To transform the center of mass coordinates to world coordinates, the following transformation is applied:

$$\begin{bmatrix} \bar{x}_\ell^W \\ \bar{y}_\ell^W \\ 1 \end{bmatrix} = \begin{bmatrix} \cos -\theta & -\sin -\theta & L_g \sin(-\theta) \\ \sin -\theta & \cos -\theta & -L_g \cos -\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{x}_\ell^{B*} \\ \bar{y}_\ell^{B*} \\ 1 \end{bmatrix}$$

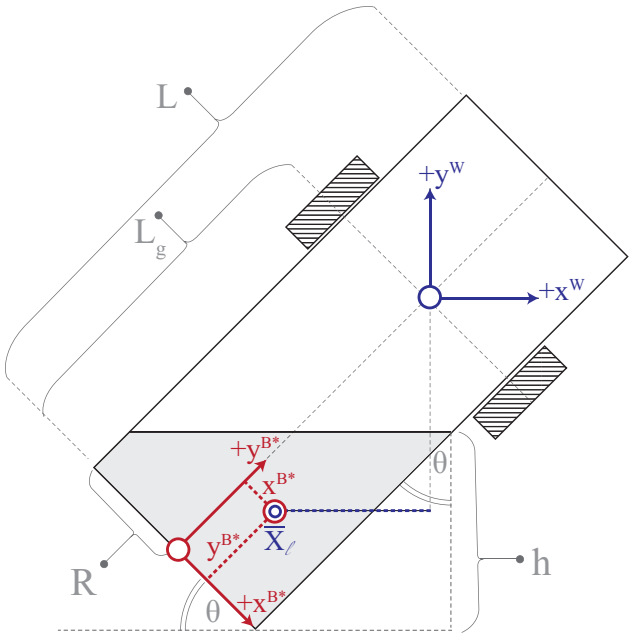


Figure A.2: An illustration of the shifted body frame $B*$, with respect to which the center of mass equations $\bar{x}_\ell^{B*}$ and $\bar{y}_\ell^{B*}$ are initially defined. $\bar{x}_\ell(\theta,V)$ is defined with respect to the world frame, so a transformation is necessary once $\bar{x}_\ell^{B*}$ and $\bar{y}_\ell^{B*}$ are computed.

and since we restrict the following definitions to $\theta \in [0, \frac{\pi}{2}]$ (symmetric reasoning leads to the derivation of the full range for theta), we can rewrite the transformation as:

$$\begin{bmatrix} \bar{x}_\ell^W \\ \bar{y}_\ell^W \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & -L_g \sin\theta \\ -\sin\theta & \cos\theta & -L_g \cos\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{x}_\ell^{B*} \\ \bar{y}_\ell^{B*} \\ 1 \end{bmatrix}$$

Thus, we can calculate the expression $\bar{x}_\ell(\theta,V)$ used in the optimization for V, which is equivalent to $\bar{x}_\ell^W$:

$$\bar{x}_\ell(\theta,V) = \bar{x}_\ell^{B*}\cos\theta + \bar{y}_\ell^{B*}\sin\theta - L_g\sin\theta$$

Case 1: Bottom end covered, top end dry

Finding the bounds for h

This case occurs when $h_{top} > h_{bottom}$ and h is within the bounds h_{bottom} and h_{top}. Geometric reasoning finds that $h_{bottom} = 2R\sin\theta$ and $h_{top} = L\cos\theta$. Thus, we have the case condition that $L\cos\theta > 2R\sin\theta$ and $h \in [2R\sin\theta, L\cos\theta]$.

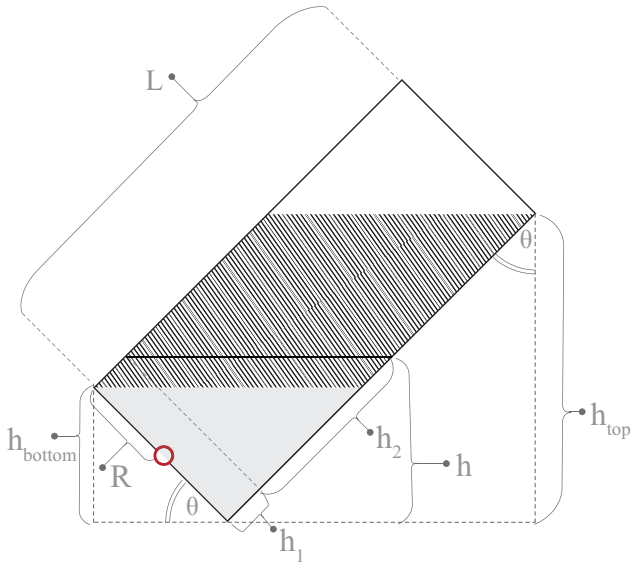


Figure A.3: Diagram depicting Case 1 for liquid center of mass and volume calculations.

Finding the expression for V

A simple way to find the expression for V for this specific case is by breaking the liquid volume into two parts, corresponding to the two values h_1 and h_2, denoted in Figure A.3.

We first find h_2 with respect to known values R and θ:

$$\begin{aligned} \tan\theta &= \frac{h_2}{2R} \\ 2R\tan\theta &= h_2 \end{aligned}$$

We can write h_1 in terms of h, R, and θ:

$$\begin{aligned} \cos\theta &= \frac{h}{h_1+h_2} \\ h_1\cos\theta + h_2\cos\theta &= h \\ h_1\cos\theta &= h - h_2\cos\theta \\ h_1 &= \frac{h}{\cos\theta} - h_2 \\ h_1 &= \frac{h}{\cos\theta} - 2R\tan\theta \end{aligned}$$

Then the volumes corresponding to h_1 and h_2 are simply $\pi R^2 h_1$ and $\frac{1}{2}\pi R^2 h_2$. In terms of h, R, and θ, the combined volume can be expressed as such:

$$V = \pi R^2\left(\frac{h}{\cos\theta} - 2R\tan\theta\right) + \pi R^3\tan\theta$$

Finding expressions for the center of mass coordinates

Calculated with respect to a shifted body frame, with the origin centered at the red ring in Figure A.3, we solve the following integrals:

$$
\begin{aligned}
\bar{x}_\ell^{B*} &= \frac{1}{M}\int\int\int x\rho dV = \frac{1}{M}\left(\frac{M}{V}\right)\int_{-R}^{R}\int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}}\int_{0}^{(R+x)\tan\theta+h_1} x dy dz dx \\
&= \frac{1}{V}\left[\int_{-R}^{R}\int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}}(R+x)\tan\theta x dz dx + \int_{-R}^{R}\int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} h_1 x dz dx\right] \\
&= \frac{1}{V}\left[2(R\tan\theta+h_1)\int_{-R}^{R} x\sqrt{R^2-x^2}dx + 2\tan\theta\int_{-R}^{R} x^2\sqrt{R^2-x^2}dx\right] \\
&= \frac{1}{V}\left[2(R\tan\theta+h_1)(-\frac{1}{3}(\cancel{R^2-x^2})^{3/2}\Big|_{-R}^{R}\right. \\
&\left. +2\tan\theta(\frac{1}{8}(x\sqrt{\cancel{R^2-x^2}}(2x^2-R^2)+R^4\arctan\frac{x}{\sqrt{\cancel{R^2-x^2}}}))\Big|_{-R}^{R}\right] \\
&= \frac{1}{V}\left[2\tan\theta\frac{1}{8}(R^4\frac{\pi}{2}--R^4\frac{\pi}{2})\right] \\
&= \frac{1}{4V}\pi R^4\tan\theta
\end{aligned}
$$

$$
\begin{aligned}
\bar{y}_\ell^{B*} &= \frac{1}{M}\int\int\int y\rho dV = \frac{1}{M}\left(\frac{M}{V}\right)\int_{-R}^{R}\int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}}\int_0^{(R+x)\tan\theta+h_1} y dy dz dx \\
&= \frac{1}{V}\frac{1}{2}\int_{-R}^{R}\int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}}((R+x)\tan\theta+h_1)^2 dz dx \\
&= \frac{1}{V}\left[\int_{-R}^{R}((R+x)^2\tan^2\theta+2(R+x)h_1\tan\theta+h_1^2)\sqrt{R^2-x^2}dx\right] \\
&= \frac{1}{V}\left[(R^2\tan^2\theta+2Rh_1\tan\theta+h_1^2)\int_{-R}^{R}\sqrt{R^2-x^2}dx\right. \\
&\left. + (2R\tan^2\theta+2h_1\tan\theta)\int_{-R}^{R}x\sqrt{R^2-x^2}dx+(\tan^2\theta)\int_{-R}^{R}x^2\sqrt{R^2-x^2}dx\right] \\
&= \frac{1}{V}\left[(R\tan\theta+h_1)^2(\frac{1}{2}(x\sqrt{\cancel{R^2-x^2}}+R^2\arctan\frac{x}{\sqrt{\cancel{R^2-x^2}}})\Big|_{-R}^{R}\right. \\
&\left. +(2R\tan^2\theta+2h_1\tan\theta)(-\frac{1}{3}(\cancel{R^2-x^2})^{3/2}\Big|_{-R}^{R})+\tan^2\theta\frac{R^4}{8}\arctan\frac{x}{\sqrt{\cancel{R^2-x^2}}}\Big|_{-R}^{R}\right] \\
&= \frac{1}{V}\left[(R\tan\theta+h_1)^2\frac{\pi R^2}{2}+\tan^2\theta\frac{\pi R^4}{8}\right] \\
&= \frac{1}{V}\left[(R\tan\theta+\frac{h}{\cos\theta}-2R\tan\theta)^2\frac{\pi R^2}{2}+\tan^2\theta\frac{\pi R^4}{8}\right]
\end{aligned}
$$

Case 2: Bottom end partially covered, top end dry

Finding the bounds for h

The bounds for h that define this case depend on the comparison between $h_{top} = L\cos\theta$ and $h_{bottom} = 2R\sin\theta$. This case occurs when $h \in [0, a]$ for $a = \min[L\cos\theta, 2R\sin\theta]$

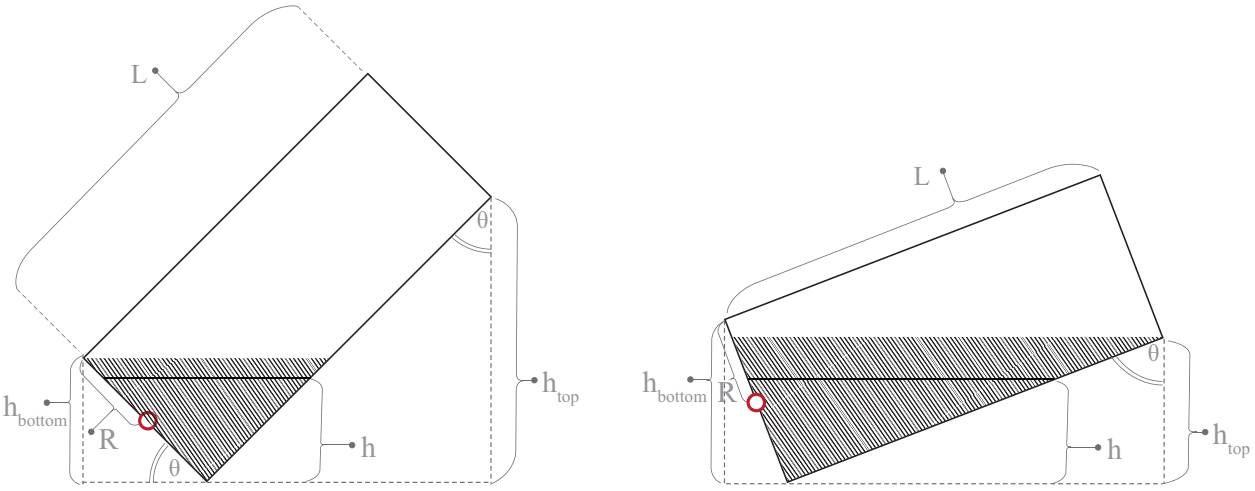


Figure A.4: Diagrams depicting Case 2 for liquid center of mass and volume calculations. (Left) for $h_{top} > h_{bottom}$. (Right) for $h_{top} < h_{bottom}$.

Finding the expression for V

To simplify our expressions, we define the following variables. Let $\alpha = R - \frac{h}{\sin\theta}$ and $\beta = \sqrt{R^2-\alpha^2}$. To find an expression for V, we evaluate the following integral:

$$
\begin{aligned}
V &= \int_{R-\frac{h}{\sin\theta}}^{R} \int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} \int_{0}^{\tan\theta(x-R+\frac{h}{\sin\theta})} dydzdx \\
&= \int_{R-\frac{h}{\sin\theta}}^{R} 2\tan\theta(x-R+\frac{h}{\sin\theta})\sqrt{R^2-x^2}dx \text{ (let } R-\frac{h}{\sin\theta}=\alpha) \\
&= -2\alpha\tan\theta\int_{\alpha}^{R}\sqrt{R^2-x^2}dx + 2\tan\theta\int_{\alpha}^{R} x\sqrt{R^2-x^2}dx \\
&= -2\alpha\tan\theta\left[\frac{1}{2}(x\sqrt{R^2-x^2}+R^2\arctan\frac{x}{\sqrt{R^2-x^2}})\Big|_{\alpha}^{R}\right] - \frac{2}{3}\tan\theta\left[(R^2-x^2)^{3/2}\Big|_{\alpha}^{R}\right] \\
&= -\alpha\tan\theta(\frac{\pi}{2}R^2 - \alpha\sqrt{R^2-\alpha^2} - R^2\arctan\frac{\alpha}{\sqrt{R^2-\alpha^2}}) + \frac{2}{3}\tan\theta(R^2-\alpha^2)^{3/2} \\
&\text{(and letting } \beta = \sqrt{R^2-\alpha^2}) \\
&= \tan\theta(\frac{2\beta^3}{3} - \alpha(\frac{\pi R^2}{2} - \alpha\beta - R^2\arctan\frac{\alpha}{\beta}))
\end{aligned}
$$

Finding expressions for the center of mass coordinates

When finding the expressions for the center of mass coordinates, we again use the simplifying variables $\alpha = R - \frac{h}{\sin\theta}$ and $\beta = \sqrt{R^2-\alpha^2}$. We evaluate the following integrals:

$$
\begin{aligned}
\bar{x}_\ell^{B*} &= \frac{1}{M}\left(\frac{M}{V}\right)\int_\alpha^R \int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} \int_0^{\tan\theta(x-\alpha)} x\,dy\,dz\,dx \\
&= \frac{1}{V}\left[\int_\alpha^R x\tan\theta(x-\alpha)2\sqrt{R^2-x^2}\,dx\right] \\
&= \frac{1}{V}\left[-2\alpha\tan\theta\int_\alpha^R x\sqrt{R^2-x^2}\,dx + 2\tan\theta\int_\alpha^R x^2\sqrt{R^2-x^2}\,dx\right] \\
&= \frac{1}{V}\left[\frac{2}{3}\alpha\tan\theta(R^2-x^2)^{3/2}\Big|_\alpha^R + \frac{1}{4}\tan\theta(x\sqrt{R^2-x^2}(2x^2-R^2) + R^4\arctan\frac{x}{\sqrt{R^2-x^2}})\Big|_\alpha^R\right] \\
&= \frac{1}{V}\left[-\frac{2}{3}\alpha\tan\theta(R^2-\alpha^2)^{3/2} + \frac{1}{4}\tan\theta(\frac{\pi}{2}R^4 - \alpha\sqrt{R^2-\alpha^2}(2\alpha^2-R^2) - R^4\arctan\frac{\alpha}{\sqrt{R^2-\alpha^2}})\right] \\
&= \frac{1}{V}\tan\theta\left[-\frac{2}{3}\alpha\beta^3 + \frac{1}{4}(\frac{\pi}{2}R^4 - \alpha\beta(2\alpha^2-R^2) - R^4\arctan\frac{\alpha}{\beta})\right]
\end{aligned}
$$

$$
\begin{aligned}
\bar{y}_\ell^{B*} &= \frac{1}{M}\left(\frac{M}{V}\right)\int_\alpha^R \int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} \int_0^{\tan\theta(x-\alpha)} y\,dy\,dz\,dx \\
&= \frac{1}{V}\int_\alpha^R \tan^2\theta(x-\alpha)^2\sqrt{R^2-x^2}\,dx \\
&= \frac{1}{V}\int_\alpha^R \tan^2\theta(x^2-2x\alpha+\alpha^2)\sqrt{R^2-x^2}\,dx \\
&= \frac{1}{V}\left[\tan^2\theta\alpha^2\int_\alpha^R \sqrt{R^2-x^2}\,dx - 2\tan^2\theta\alpha\int_\alpha^R x\sqrt{R^2-x^2}\,dx + \tan^2\theta\int_\alpha^R x^2\sqrt{R^2-x^2}\,dx\right] \\
&= \frac{1}{V}\left[\frac{1}{2}\tan^2\theta\alpha^2(x\sqrt{R^2-x^2} + R^2\arctan\frac{x}{\sqrt{R^2-x^2}})\Big|_\alpha^R + \frac{2}{3}\tan^2\theta\alpha((R^2-x^2)^{3/2})\Big|_\alpha^R\right. \\
&\left. + \frac{1}{8}\tan^2\theta(x\sqrt{R^2-x^2}(2x^2-R^2) + R^4\arctan\frac{x}{\sqrt{R^2-x^2}})\Big|_\alpha^R\right] \\
&= \frac{1}{V}\left[\frac{1}{2}\tan^2\theta\alpha^2(\frac{\pi}{2}R^2 - \alpha\sqrt{R^2-\alpha^2} - R^2\arctan\frac{\alpha}{\sqrt{R^2-\alpha^2}}) - \frac{2}{3}\tan^2\theta\alpha(R^2-\alpha^2)^{3/2}\right. \\
&\left. + \frac{1}{8}\tan^2\theta(\frac{\pi}{2}R^4 - \alpha\sqrt{R^2-\alpha^2}(2\alpha^2-R^2) - R^4\arctan\frac{\alpha}{\sqrt{R^2-\alpha^2}})\right] \\
&= \frac{1}{V}\tan^2\theta\left[\frac{1}{2}\alpha^2(\frac{\pi}{2}R^2 - \alpha\beta - R^2\arctan\frac{\alpha}{\beta}) - \frac{2}{3}\alpha\beta^3 + \frac{1}{8}(\frac{\pi}{2}R^4 - \alpha\beta(2\alpha^2-R^2) - R^4\arctan\frac{\alpha}{\beta})\right]
\end{aligned}
$$

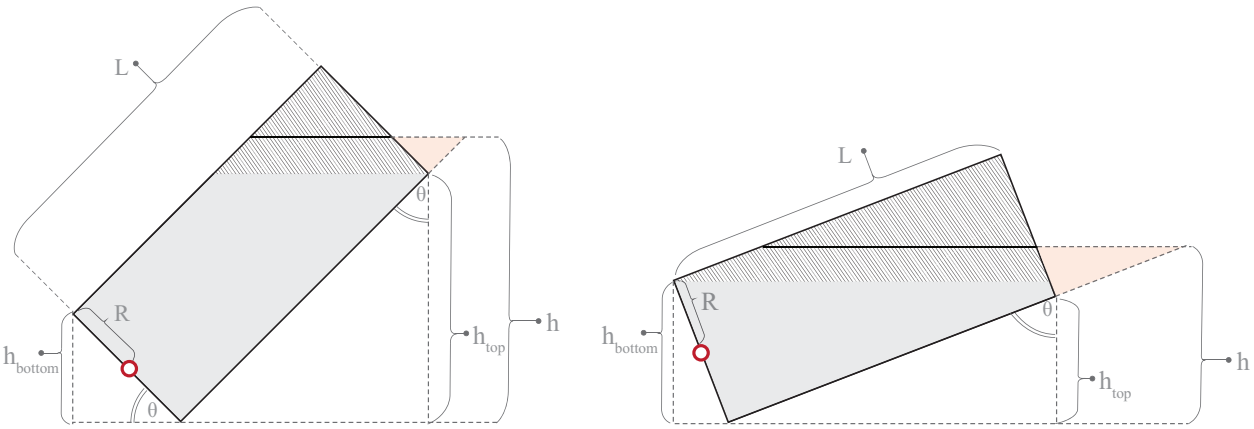


Figure A.5: Diagrams depicting Case 3 for liquid center of mass and volume calculations. (Left) for $h_{top} > h_{bottom}$. (Right) for $h_{top} < h_{bottom}$. Red triangles correspond to volume (case 2) with height $h - L\cos\theta$ to be subtracted from volume (case 1) with height h.

Case 3

Finding the bounds for h

For this specific case, the bounds for h again rely on a comparison between $h_{top} = L\cos\theta$ and $h_{bottom} = 2R\sin\theta$. This case occurs when $h \in [\alpha, L\cos\theta + 2R\sin\theta]$ for $\alpha = \max[L\cos\theta, 2R\sin\theta]$.

Finding the expression for V

The resulting cylindrical wedge of this specific case can be simply calculated as the volumetric difference of a Case 1 wedge with height $h_1 = h$ and a Case 2 wedge with height $h_2 = h - L\cos\theta$. A cross-section of the Case 2 wedge is depicted in red in Figure A.5. If we let $V_i(\cdot)$ denote the volume formula for case (i) as a function of height $(\cdot)$, then the volume of the cylindrical wedge can be calculated as:

$$V_3(h) = V_1(h_1) - V_2(h_2)$$

Finding the expressions for the center of mass coordinates

Additionally, let $\bar{x}^B_{\ell_i}(\cdot)$ and $\bar{y}^B_{\ell_i}(\cdot)$ denote the center of mass formulas for case (i) as functions of height $(\cdot)$. Note that, since the centers of mass are calculated in reference to the shifted body frame, with the origin centered at the red dot in Figure A.5, the values for $\bar{y}^B_{\ell_2}(\cdot)$ will be shifted by the length of the container, L. The center of mass coordinates, with respect to the shifted body frame, are thus calculated as such:

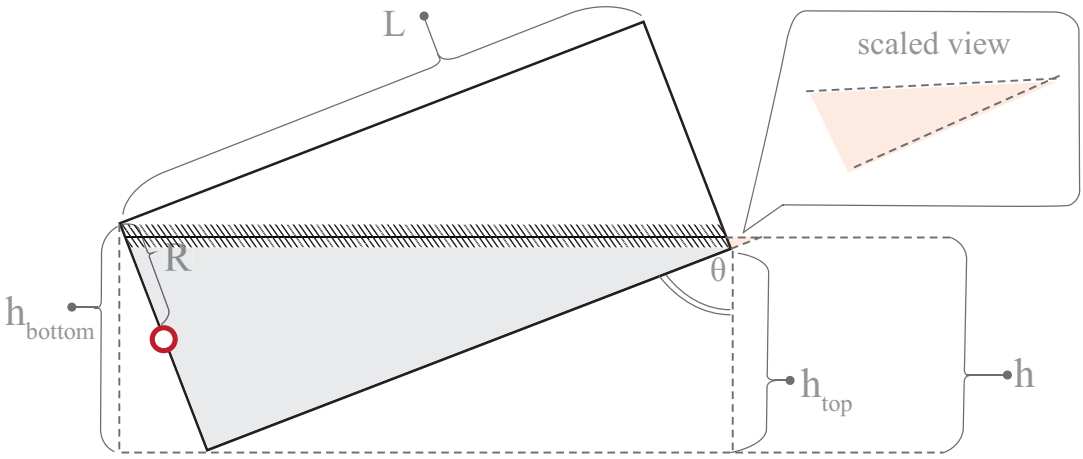


Figure A.6: Diagram depicting Case 4 for liquid center of mass and volume calculations. Red triangle corresponds to volume (case 2) with height $h - L\cos\theta$ to be subtracted from volume (case 2) with height h.

$$\bar{x}^B_{\ell_3} = \frac{\bar{x}^B_{\ell_1}(h_1)V_1(h_1) - \bar{x}^B_{\ell_2}(h_2)V_2(h_2)}{V_1(h_1) - V_2(h_2)}$$
$$\bar{y}^B_{\ell_3} = \frac{\bar{y}^B_{\ell_1}(h_1)V_1(h_1) - (\bar{y}^B_{\ell_2}(h_2) + L)V_2(h_2)}{V_1(h_1) - V_2(h_2)}$$

Case 4

Finding the bounds for h

This specific case only occurs when $L\cos\theta < 2R\sin\theta$ and $h \in [L\cos\theta, 2R\sin\theta]$.

Finding the expression for V

The resulting cylindrical wedge of this specific case can be simply calculated as the volumetric difference of a Case 2 wedge with height $h_1 = h$ and a Case 2 wedge with height $h_2 = h - L\cos\theta$. A cross-section of the subtracted Case 2 wedge is depicted in red in Figure A.5. As before, let $V_i(\cdot)$ denote the volume formula for case (i) as a function of height $(\cdot)$. The volume of the cylindrical wedge can be calculated as:

$$V_4(h) = V_2(h_1) - V_2(h_2)$$

Finding the expressions for the center of mass coordinates

As before, let $\bar{x}^B_{\ell_i}(\cdot)$ and $\bar{y}^B_{\ell_i}(\cdot)$ denote the center of mass formulas for case (i) as functions of height $(\cdot)$. Note that, since the centers of mass are calculated in reference to the shifted body frame, with the origin centered at the red dot in Figure A.5, the values for $\bar{y}^B_{\ell_2}(\cdot)$ will be shifted by the length of the container, L. The center of mass coordinates, with respect to the shifted body frame, are thus calculated as such:

$$\bar{x}^B_{\ell_4} = \frac{\bar{x}^B_{\ell_2}(h_1)V_2(h_1) - \bar{x}^B_{\ell_2}(h_2)V_2(h_2)}{V_2(h_1) - V_2(h_2)}$$
$$\bar{y}^B_{\ell_4} = \frac{\bar{y}^B_{\ell_2}(h_1)V_2(h_1) - (\bar{y}^B_{\ell_2}(h_2) + L)V_2(h_2)}{V_2(h_1) - V_2(h_2)}$$

A.3 Liquid Viscosity

Low-viscous fluids

By inducing sloshing and observing the slosh-induced forces and torques caused by the shifting center of mass of the ever-evolving liquid volume, we can estimate the viscosity of the liquid. A decaying oscillatory wave in either force or torque can be used to estimate this value. As noted in Section 2.4, the authors of [43, 138, 75] define the following empirical model. Let

$$\Delta = \frac{\text{peak amplitude of oscillation}}{\text{peak amplitude of oscillation one cycle later}}$$

be the logarithmic decrement of the decaying oscillation. The damping ratio is then defined by:

$$\gamma = \frac{\Delta}{\sqrt{2\pi^2 + \Delta^2}}$$

For an upright circular cylindrical tank, the damping ratio of the first symmetric mode is given by:

$$\gamma = 0.79\sqrt{\frac{\nu}{R^{3/2}g^{1/2}}}\left[1 + \frac{0.318}{\sinh 1.84h/R}\left(\frac{1-(h/R)}{\cosh 1.84h/R} + 1\right)\right]$$

where ν is the kinematic viscosity of the liquid, R is the radius of the tank, and h is the fill-level of the tank. We want an expression for dynamic viscosity, $\mu = \nu\rho$. Thus, we first rearrange the above equation.

$$\gamma = 0.79\sqrt{\frac{\nu}{R^{3/2}g^{1/2}}}\left[1 + \frac{0.318}{\sinh 1.84h/R}\left(\frac{1-(h/R)}{\cosh 1.84h/R} + 1\right)\right]$$
$$\gamma^2 = \frac{\nu}{R^{3/2}g^{1/2}}0.79^2\left[1 + \frac{0.318}{\sinh 1.84h/R}\left(\frac{1-(h/R)}{\cosh 1.84h/R} + 1\right)\right]^2$$
$$\nu = \gamma^2\sqrt{R^3 g}\left[0.79\left(1 + \frac{0.318}{\sinh 1.84h/R}\left(\frac{1-(h/R)}{\cosh 1.84h/R} + 1\right)\right)\right]^{-2}$$

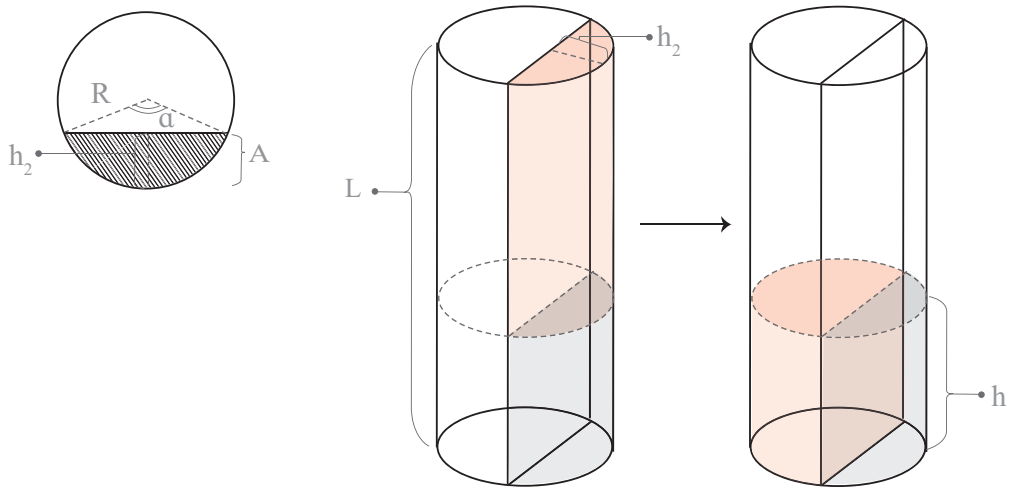


Figure A.7: (Left): Cross-section of liquid flow; (Middle): Highlighted regions (red+grey) are to denote where liquid is present immediately after the rotation of the container. Liquid flows through the cross-sectional area illustrated on the left. (Right): Highlighted regions (red+grey) are to denote where liquid is present after reaching steady state. The red regions are to illustrate the change in location of mass before and after liquid flow.

Plugging in the expression for γ and multiplying by ρ, the density of the liquid, we get an expression for the dynamic viscosity μ:

$$\mu = \rho \left(\frac{\Delta}{\sqrt{2\pi^2 + \Delta^2}} \right)^2 \sqrt{R^3 g} \left[0.79 \left(1 + \frac{0.318}{\sinh 1.84 h/R} \left(\frac{1-(h/R)}{\cosh 1.84 h/R} + 1 \right) \right) \right]^{-2}$$

High-viscous fluids

Instead of observing sloshing dynamics, the viscosity for high-viscous fluids can be estimated by measuring the rise-time of the overdamped torque response induced by rotating the container. Specifically, the approximate time it takes for the liquid to settle from and to the states illustrated in the middle and right diagrams of Figure A.7 can be used to calculate mass flux, which relates to dynamic viscosity μ.

Mass flux $\mathcal{Q}$ can be approximated by $\frac{\Delta m_\ell}{t_r A}$, where t_r is the measured 10-90% rise time of the damped torque response, Δm_ℓ is the change in mass over that time t_r (corresponding to the red region in Figure A.7, and A is the cross-sectional area through which the mass flows. As shown in Figure A.7, A is the circular segment with height h_2. The formula for this cross-sectional area is given by: $A = \frac{R^2}{2}(\alpha - \sin\alpha)$, where $\alpha = 2\arccos\frac{R-h_2}{R}$. With an estimated volume V, h_2 can be found by searching for the height of the liquid at steady state when the container is tilted by 90 deg. Finally, letting h be the height of the liquid at steady state when the container is tilted by 0 deg, shown on the right in Figure A.7, we can approximate Δm_ℓ by the mass quantity estimated to move

during fluid flow induced by the rotation. This corresponds to the mass of the region highlighted in red. We can find Δm_ℓ by multiplying the total mass of the liquid, m_ℓ, by the ratio of liquid volume that changes location before and after fluid flow. The volume of the red region can be calculated as: $V - Ah$. Thus, $\Delta m_\ell = m_\ell(\frac{V-Ah}{V})$.

For a free surface flow on an inclined plane, mass flux is defined as: $\mathcal{Q} = \frac{\rho g \cos\theta h_2^3}{2\mu}$. Thus, we can rearrange this equation to find μ: $\mu = \frac{\rho g \cos\theta h_2^3}{2\mathcal{Q}}$. Plugging in our approximation of $\mathcal{Q}$ and $\theta = 0$, we get:

$$\mu = \frac{\rho g h_2^3 V t_r A}{2 m_\ell (V - Ah)}$$

Appendix B

Framework Details for Chapter 4

B.1 Feature vector X

Feature vector X is composed of four elements. Let X_t refer to the feature in X relevant to time and X_p refer to the three features relevant to position. Below, we derive the relationship between the coefficient of restitution e and feature X_t, as well as how the features in X_p are extracted from the positions of the first three ball bounces.

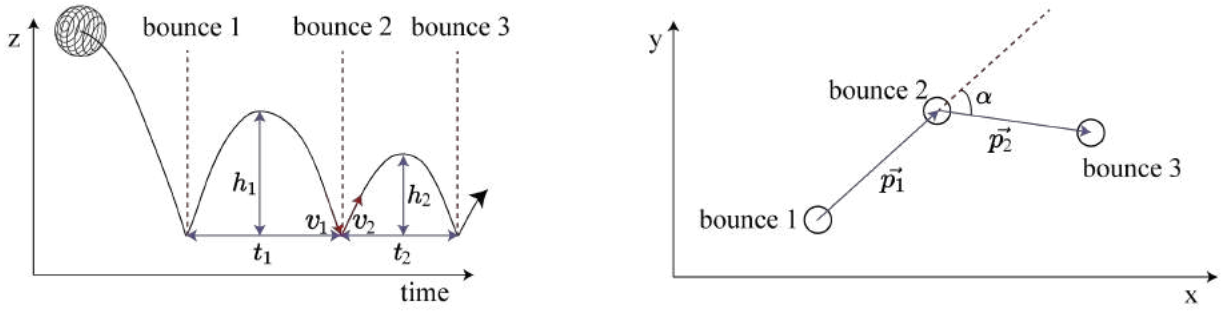


Figure B.1: (Left): A graph of the vertical trajectory of a ball over time, labeled with relevant variables for extracting X_t. (Right): xy planar positions of ball bounces are denoted by the open circles, and relevant variables for extracting X_p are labeled.

Derivation of X_t from e

The coefficient of restitution e is an expression relating the relative kinetic energy of a system before and after a collision. For a bouncing ball on a perfectly rigid surface, e can be expressed by the following ratio:

$$e = \sqrt{\frac{\text{KE}_{\text{after collision}}}{\text{KE}_{\text{before collision}}}} \tag{B.1}$$

We would like to extract e from audio signals of a bouncing ball, assuming that it is dropped with 0 initial linear velocity. With this assumption, the problem can be simplified to 1-dimension, where the ball travels only in the vertical direction (this is, of course, assuming for now that the ball is not perturbed upon collision). Letting v_1 and v_2 be the ball velocity before and after the 2nd bounce (see Figure B.1), then e can be rewritten in terms of the entering and exiting velocities:

$$e = \sqrt{\frac{1/2mv_2^2}{1/2mv_1^2}} = \frac{v_2}{v_1} \tag{B.2}$$

However, velocities are difficult to measure or perceive with sound. Thus, we want to relate e to the times between bounces 1 and 2 (t_1) and bounces 2 and 3 (t_2). Let h_1 and h_2 be defined by the heights depicted in Figure B.1. Assuming that, between collisions, the bouncing ball trajectory can be defined by classical projectile equations (assuming an absence of factors such as air resistance), then v_1 can be related to t_1 by the following derivation:

We know from classical projectile equations that:

$$\frac{1}{2}mv_1^2 = mgh_1 \text{ and } \frac{1}{2}g(\frac{t_1}{2})^2 = h_1 \tag{B.3}$$

Then combining the above equations, we get a relationship for v_1 and t_1:

$$\frac{1}{2}v_1^2 = \frac{1}{2}g^2(\frac{t_1}{2})^2 \rightarrow v_1 = \frac{gt_1}{2} \tag{B.4}$$

Thus, e can be rewritten in terms of the times between bounces:

$$e = \frac{v_2}{v_1} = \frac{t_2}{t_1} \tag{B.5}$$

Because we wanted to infer the parameter e, we chose $X_t = \frac{t_2}{t_1}$

In reality, perturbations occur at each bounce due to factors such as surface asperities. Consequently, the velocity of the ball has both horizontal and vertical components. Letting d_1 and d_2 be defined by the horizontal distances between bounces 1 and 2 and bounces 2 and 3, respectively, then e would actually be derived from the ratio in Equation B.2 as:

$$e = \frac{v_2}{v_1} = \frac{\sqrt{v_{2_{\text{horizontal}}}^2 + v_{2_{\text{vertical}}}^2}}{\sqrt{v_{1_{\text{horizontal}}}^2 + v_{1_{\text{vertical}}}^2}} = \sqrt{\frac{(d_2/t_2)^2 + g^2(t_2/2)^2}{(d_1/t_1)^2 + g^2(t_1/2)^2}} \tag{B.6}$$

However, since we capture positional information in the last three features of X, we decided to separate the temporal information from the positional information. Thus, we chose the simpler ratio of e for X_t, which only depended on t_1 and t_2.

Extracting X_p from ball bounce locations

The last three features of X, or X_p, encode positional information regarding the first three bounces. Letting $\vec{p}_1$ be the positional vector from the location of bounce 1 to the location of bounce 2, and $\vec{p}_2$ be the positional vector from the location of bounce 2 to the location of bounce 3 (see Figure B.1), then the distances between the bounces can be calculated as:

$$d_1 = ||\vec{p}_1||_2 \text{ and } d_2 = ||\vec{p}_2||_2 \tag{B.7}$$

We use d_1 and d_2 as the second and third features of X. Combined with the temporal information in X_t, a more accurate estimation of e can be inferred.

The last feature of X_p encodes directional perturbations upon collision. Specifically, we find the signed angle α between $\vec{p}_1$ and $\vec{p}_2$ via the following expression:

$$\alpha = \text{sign}(\frac{\vec{p}_{1_x}\vec{p}_{2_y} - \vec{p}_{2_x}\vec{p}_{1_y}}{||\vec{p}_1||||\vec{p}_2||}) * \arccos(\frac{\vec{p}_1}{||\vec{p}_1||} \cdot \frac{\vec{p}_2}{||\vec{p}_2||}) \tag{B.8}$$

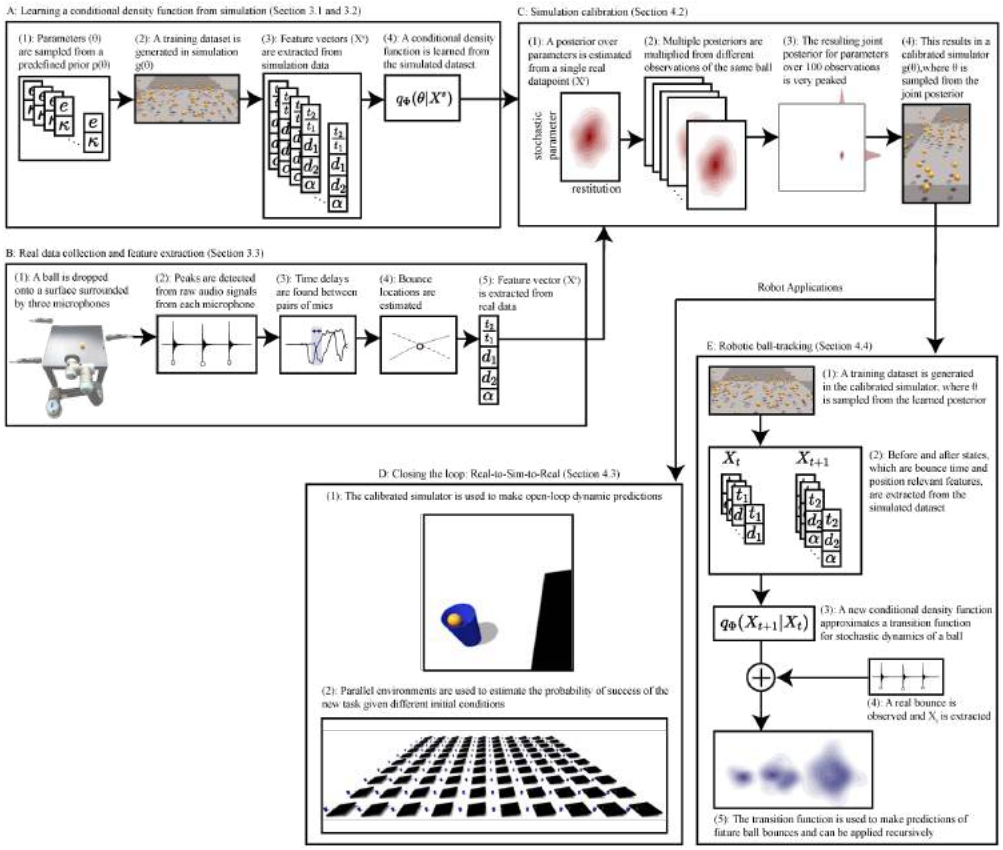


Figure B.2: A flow diagram detailing the STReSSD framework. (A): Simulation is used to learn the conditional density function $q_\Phi(\theta|X^s)$ relating parameters e and κ to observation features $X = [\frac{t_2}{t_1}, d_1, d_2, \alpha]$. 6000 independent simulations are run to generate the training set to learn q_Φ. (B): Feature extraction of real data begins by observing the peaks of raw audio signals, finding the time delays between pairs of microphones, and locating the bounces. The low-dimensional feature vector X^r is thus extracted from raw data. (C): The simulator is calibrated using real observations to match a particular ball's dynamic behavior. A posterior over the parameters $\theta = [e, \kappa]$ is approximated using the learned conditional density function q_Φ and the real observation X^r. Posteriors from different observations of the same ball can be multiplied to generate a more peaked joint posterior. The calibrated simulator samples from this joint posterior. (D): The calibrated simulator from (C) is used to make open-loop dynamic predictions of the ball. (E): The calibrated simulator is used for robotic ball-tracking. A new training set is generated by forward simulating the calibrated simulator, with relevant parameters sampled from the joint posterior in (C). New features ($X_t = [t_t, d_t]$ and $X_{t+1} = [t_{t+1}, d_{t+1}, \alpha_{t+1}]$) are extracted from this simulator and a second conditional density function approximates a transition function for the stochastic dynamics of the ball. The transition function is used to estimate the next bounce location and time, given the last two bounces. The transition function can be applied recursively to predict future ball bounces.

B.2 Sound localization

In this section, we derive the Time Delay Estimation equations for 2D-localization using 3 microphones and compare the accuracy of offline and real-time sound localization methods.

Time Delay Estimation with 3 Microphones

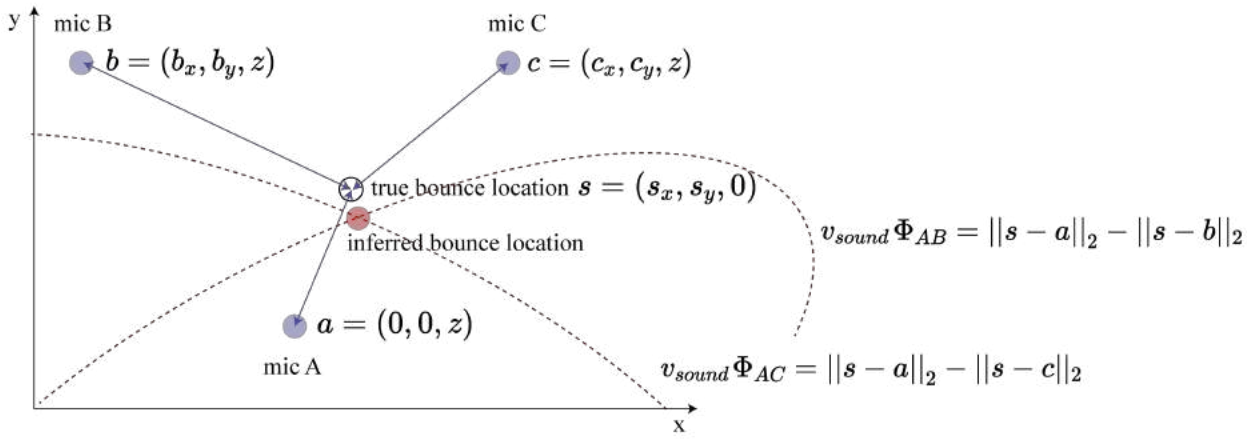


Figure B.3: Locating the source of a sound using time delay estimation involves finding the intersection of two conics, defined by the time delays between pairs of microphones.

Let v_{sound} be the speed of sound through air, Φ_{AB} and Φ_{AC} be the time delays between signals received at microphones A and B and A and C, and denote the positions of the ball strike and microphones as $s = (s_x, s_y, 0)$, $a = (0, 0, z)$, $b = (b_x, b_y, z)$, and $c = (c_x, c_y, z)$, respectively. Then the time delay Φ_{AB} is a function of the relative distances of the microphones to the ball strike, or

$$\Phi_{AB} = \frac{1}{v_{sound}}(||s-a||_2 - ||s-b||_2) \tag{B.9}$$

Thus, solving for the location of the ball strike is equivalent to finding the intersection of two conics (see Figure B.3):

$$v_{sound}\Phi_{AB} = ||s-a||_2 - ||s-b||_2 \text{ and } v_{sound}\Phi_{AC} = ||s-a||_2 - ||s-c||_2 \tag{B.10}$$

A nonlinear optimizer is used to find this intersection. Because there can be more than one solution, we initialize the optimizer with a location inside the convex hull of the microphone positions.

Comparison of Offline and Real-Time Sound Localization

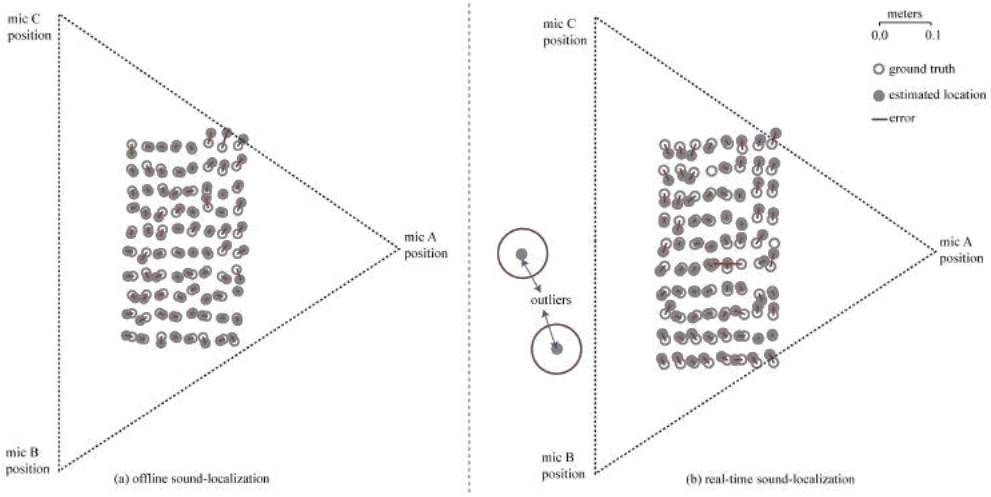


Figure B.4: A comparison of (a) offline and (b) online sound localization. Ground truth positions were recorded by tapping a piece of paper layered with a carbon sheet. These positions were converted to world coordinates via an automated image processing script. Observe that outliers occasionally arise when using the less-accurate real-time sound localization method. However, the displacement errors of the outliers are sufficiently large such that they can be easily filtered.

B.3 Dynamic Predictions without Collision Noise

Below, we supplement the description of dynamic predictions without collision noise in Section 4.3 with relevant equations. The robot control sequence is summarized in Algorithm 1.

Let (x_i, y_i), t_i, and d_i be the position of the most recent bounce, time, and distance between the most recent and prior bounces (see Figure B.5). Assuming that the ball travels in a piecewise parabolic trajectory with no directional perturbations upon each bounce, we can estimate the exit velocity of the ball rebounding from the bounce to be defined by:

$$v_{e(xy_i)} = \frac{ed_i}{t_i} \text{ and } v_{e(z_i)} = \frac{egt_i}{2} \tag{B.11}$$

where g is the gravitational constant, the xy subscript refers to the horizontal velocity of the ball, and the z subscript refers to the vertical velocity component of the ball. These estimates allow us to calculate whether or not the ball will intersect the plane before the next bounce. First, denote d_R as the horizontal distance from the most recent bounce to the robot plane along the direction of travel (see Figure B.5). The distance and time of the next bounce relative to the most recent bounce, or d_{i+1} and t_{i+1}, can be calculated by:

$$d_{i+1} = e^2 d_i \text{ and } t_{i+1} = et_i \tag{B.12}$$

Then to calculate whether or not the ball will intersect the robot plane before the next bounce, we check if

$$d_{i+1} > d_R \tag{B.13}$$

If true, the ball will intersect the plane before it lands again in t_R seconds, so the end-effector must be moved to the intersection point (x_R, y_R, z_R) in t_R seconds. If the ball will not intersect the plane before it lands again, the end-effector moves to y_R in t_{i+1} seconds before the next bounce is observed.

The intersection point (x_R, y_R, z_R) is partially pre-determined and therefore fixed by the placement of the robot (which sits at $x = x_R$). We can calculate y_R by finding the intersection between line $x = x_R$ and the line connecting the last two ball bounce positions (see Figure B.5). This intersection occurs at:

$$y_R = \frac{y_i - y_{i-1}}{x_i - x_{i-1}} x_R + (y_i - \frac{y_i - y_{i-1}}{x_i - x_{i-1}} x_i) \tag{B.14}$$

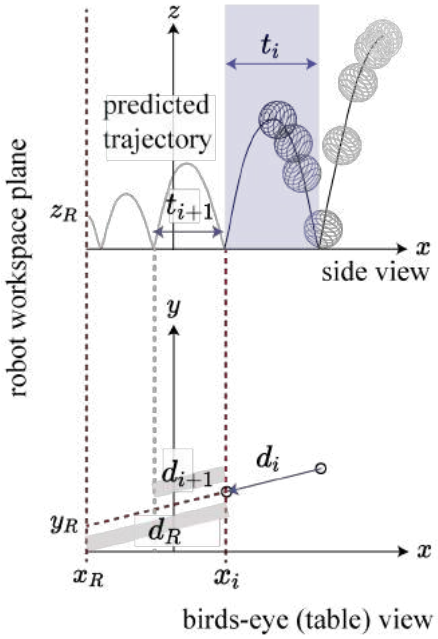


Figure B.5: Dynamic predictions without collision noise assume a piecewise parabolic trajectory for the ball, based on the most recent two bounces. The robot moves to the intersection of this predicted trajectory and the robot workspace plane.

Finally, if it is determined that the ball will intersect the robot workspace plane before it bounces again, classical projectile equations can be used to calculate t_R and z_R:

$$t_R = \frac{d_R}{v_{e_{xy_i}}} = \frac{d_R t_i}{e d_i} \tag{B.15}$$

$$z_R = v_{e(z_i)} t_R - \frac{1}{2} g t_R^2 = (\frac{egt_i}{2})(\frac{d_R t_i}{ed_i}) - \frac{1}{2} g (\frac{d_R t_i}{ed_i})^2 \tag{B.16}$$

Algorithm 1: Robot Control using Dynamic Predictions without Collision Noise

Result: Positions the end-effector of a robotic arm to meet a bouncing ball

while *Received bounce position and times* **do**
  $(x, y) \leftarrow$ bounce location;
  if *First bounce* **then**
    Move end-effector to position y;
  else
    if *After next bounce, the ball will intersect the robot plane at* (y_R, z_R) **then**
      Move end-effector to position (y_R, z_R) in t_R seconds;
    else
      Move end-effector to y_R in t_{i+1} seconds (before the next bounce is observed);
    end
  end
end

B.4 Dynamic Predictions with Collision Noise

In this section, the algorithm for making dynamic predictions with collision noise is outlined in Algorithm 2. Additionally, a complete breakdown of all 30 tosses per ball is included, and the failure modes due to outliers are compared between Algorithm 1 and Algorithm 2. Finally, two real examples are shown with freeze-frames in Section 15 to detail the algorithmic steps associated with each bounce event.

Algorithm 2: Robot Control using Dynamic Predictions with Collision Noise

Result: Positions the end-effector of a robotic arm to meet a bouncing ball

1 **while** *Received bounce position and times* **do**
2 $(x, y) \leftarrow$ bounce location;
3 **if** *First bounce* **then**
4 Move end-effector to position y;
5 **else if** *Second bounce* **then**
6 next_bounce_posterior $\leftarrow$ Generate posterior for next bounce position and time ;
7 Look ahead k bounces to generate n samples where the ball will intersect the plane ;
8 **if** *Any samples intersect the plane of the robot* **then**
9 Move $\propto \sigma^{-1}$ distance toward mean of samples, where σ is the standard deviation of the intersection points on the robot plane ;
10 **else**
11 **if** *Bounce position is an outlier* **then**
12 Replace bounce location with the mean of next_bounce_posterior ;
13 Repeat lines 6-9 ;
14 **end**
15 **end**

As discussed in Section 4.3, a new BayesSim model is trained to generate posteriors over the next bounce time and location, given the time and distance between the prior two bounces. After the second bounce is observed, this new BayesSim model can be used to generate a posterior for the next bounce position and time. Furthermore, the model can be used to recursively sample future bounce positions until they pass the robot workspace plane. For example, let k be the number of bounces to look ahead and n be the number of samples to generate at each future bounce. Given the last two bounces i and $i-1$, a posterior over the future bounce $i+1$ is generated using the BayesSim model. This posterior is sampled n times, and each sample can be used to create a new posterior with the most recent bounce i. Thus, n posteriors over bounce $i+1$ are generated. This is repeated up to k times until the sample intersects the workspace plane. A maximum of n^k samples are generated in this step. In experiments, we found that $n = 10$ and $k = [2,4]$ were sufficient at capturing the uncertainty in future bounce locations and did not significantly add to the latency of the entire algorithm (this inference step ran at about 20 Hz). We varied k depending on how reactive we wanted the robot to be. For instance, the moonball is highly uncertain at each collision, so we used $k = 2$. In contrast, since the ping-pong ball experienced lower collision noise, we used $k = 4$.

At every subsequent bounce after the second bounce, the bounce location is classified as either an outlier or an inlier by using an elliptic envelope filter with the future bounce posterior generated at the prior bounce. If the bounce location is determined to be an outlier, then the mean of that posterior is used in place of the outlier location. This extra check was implemented to compensate for inaccurate real-time bounce localization due to high reverberation. In the following section, we compare the sensitivity to outliers using Algorithms 1 and 2.

Full Breakdown of Success and Failures out of all 30 Tosses

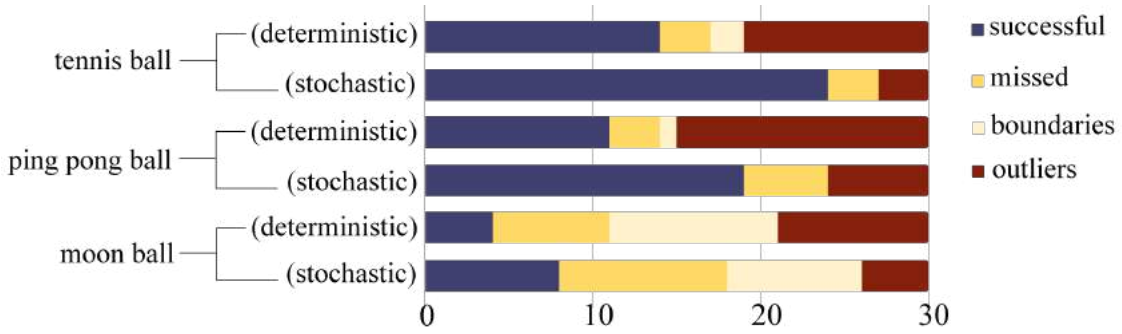


Figure B.6: ”Outliers” signifies failures due to errors in sound localization, ”Boundaries” refers to failures due to the ball violating robot workspace boundaries, and ”Missed” denotes any other failure. All 30 tosses are represented in this graph.

Section 4.4 performs analysis on the tosses that do not violate the robot's workspace constraints. For completeness, we include a full breakdown of all 30 tosses in Figure B.6, which includes the failures due to the ball violating the robot workspace boundaries. As shown in Figure B.6, few to none of the tosses violate the boundary for the tennis ball and ping pong ball, but almost a third of the tosses for the moon ball result in boundary violations. This is because the moonball experiences highly stochastic collisions.

Comparison of Sensitivity to Outliers with Algorithms 1 and 2

In Figure B.7, we break down the failure modes of both Algorithms 1 and 2 due to outliers in sound localization. As shown in the graph, both methods are sensitive to outliers in the first or second bounce. Because Algorithm 2 relies on accurate first and second bounce locations to predict future bounce distributions, there currently is not a way to adjust for these errors. We believe that we can eliminate these outliers by implementing more efficient real-time processing, for instance, by using a real-time audio-processing language called SuperCollider, or by training a neural network to perform fast peak detection. However, by considering collision noise for dynamic predictions, Figure B.7 shows that Algorithm 2 successfully filters out outliers in any subsequent bounces and can compensate for their errors.

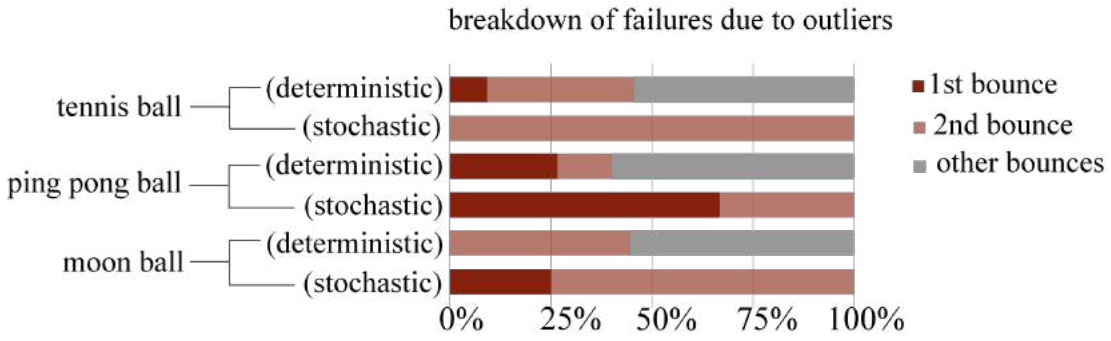


Figure B.7: We categorize the failures due to outliers in bounce localization by the bounce at which the outlier occurs. Note that Algorithm 2 successfully filters out outliers as long as they occur after the second bounce. The robot is able to use, instead, the mean of the posterior over which the bounce *should* have occurred, which subsequently leads to more accurate ball trajectory predictions.

Example robot trials with captions

Below, we detail two real robotic examples with freeze-frames to depict the steps of Algorithm 2.
Example 1: Success with no outliers

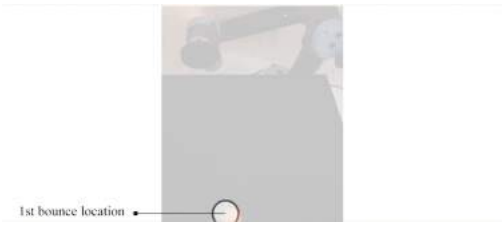


Figure B.8: The first bounce is located.

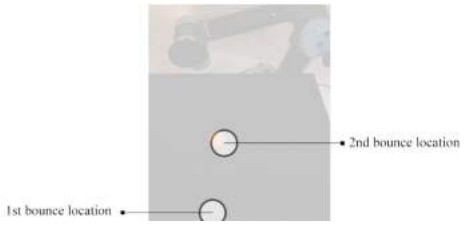


Figure B.9: The second bounce is located.

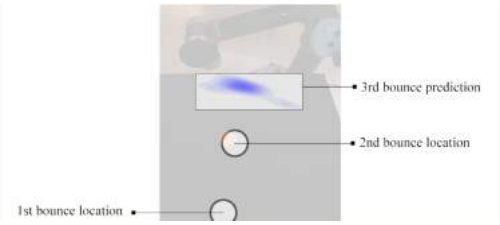


Figure B.10: The 1st and 2nd measured locations are used to predict the distribution over the 3rd bounce location.

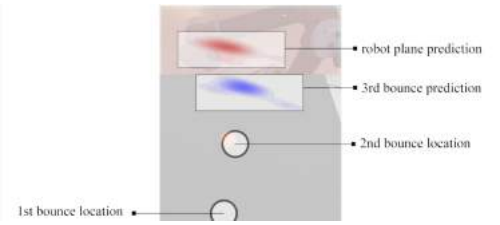


Figure B.11: The 2nd measured location and the 3rd bounce location distribution are used to predict the distribution of the robot plane intersections.

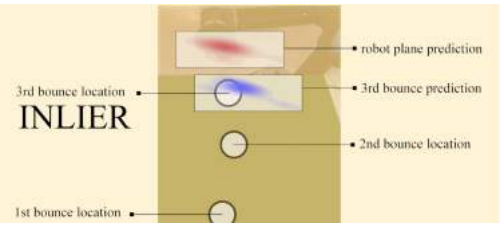


Figure B.12: Inlier: the audio-based localization of the 3rd bounce lies within the predicted distribution and is trusted. This leads to a successful prediction of the robot plane intersection, and the robot comes into contact with the ball

Example 2: Success with an outlier

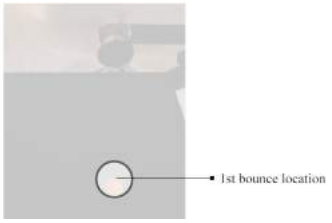


Figure B.13: The first bounce is located.

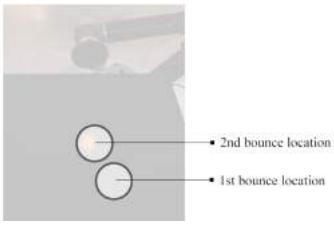


Figure B.14: The second bounce is located.

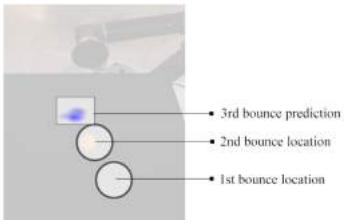


Figure B.15: The 1st and 2nd measured locations are used to predict the distribution over the 3rd bounce location.

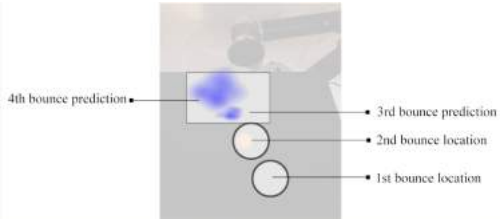


Figure B.16: The 2nd measured location and 3rd bounce location distribution are used to predict the distribution over the 4th bounce location.

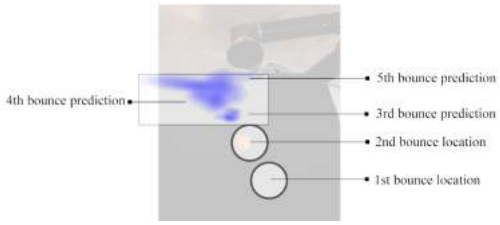


Figure B.17: The 3rd and 4th bounce location distributions are used to predict the distribution over the 5th bounce location.

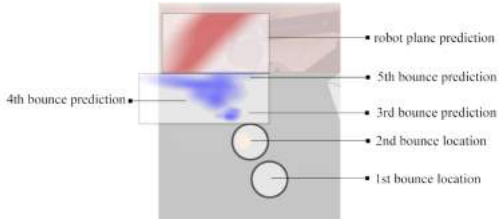


Figure B.18: The 4th and 5th bounce location distributions are used to predict the distribution of the robot plane intersections.

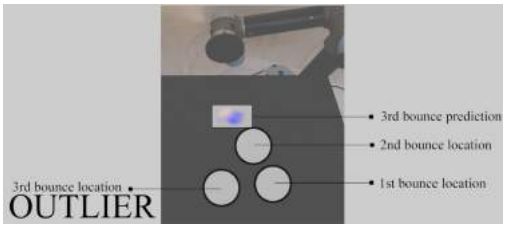


Figure B.19: Outlier: the audio-based localization of the 3rd bounce lies far from the predicted distribution and is ignored.

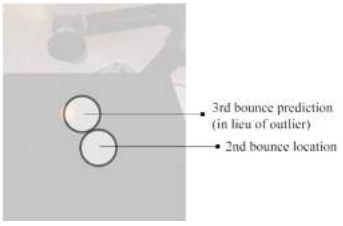


Figure B.20: The mean of the predicted distribution for the 3rd bounce is used in lieu of the outlier.

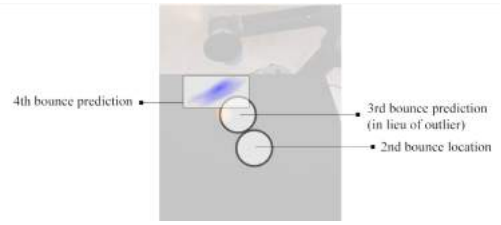


Figure B.21: The 2nd measured location, as well as the 3rd *predicted* location, are used to predict the distribution of the 4th bounce.

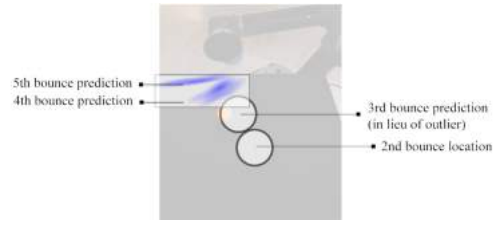


Figure B.22: The 3rd *predicted* location, as well as the 4th bounce location distribution, are used to predict the distribution of the 5th bounce.

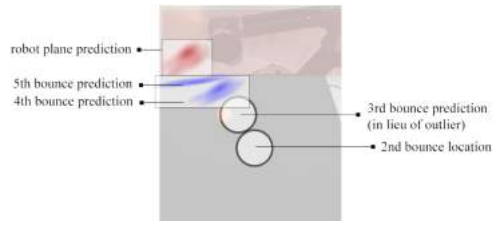


Figure B.23: The 4th and 5th bounce location distributions are used to predict the distribution of the robot plane intersections.

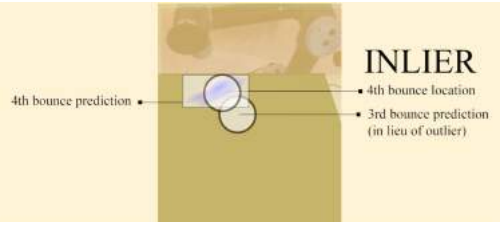


Figure B.24: Inlier: the audio-based localization of the 4th bounce lies within the predicted distribution.

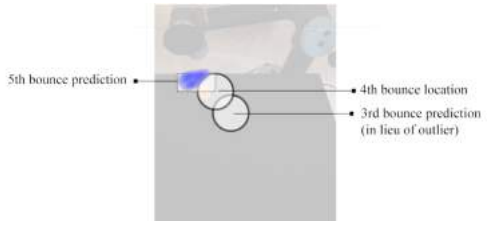


Figure B.25: The 3rd *predicted* location, as well as the 4th measured location, are used to predict the distribution of the 5th bounce.

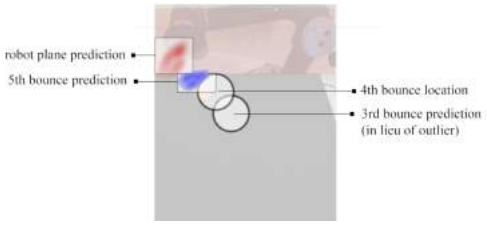


Figure B.26: The 4th measured location and the distribution of the 5th bounce are used to predict the distribution of the robot plane intersections.

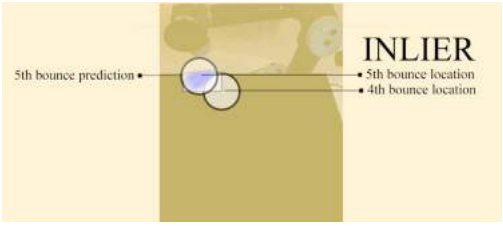


Figure B.27: Inlier: the audio-based localization of the 5th bounce lies within the predicted distribution.

Figure B.28: Successful prediction