

Interactive Touch for Manipulation

by

Shaoxiong Wang

B.E., Tsinghua University (2017)

S.M., Massachusetts Institute of Technology (2019)

Submitted to the Department of Electrical Engineering and Computer
Science

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2022

© Massachusetts Institute of Technology 2022. All rights reserved.

Author
Department of Electrical Engineering and Computer Science
July 15, 2022

Certified by.....
Edward H. Adelson
John and Dorothy Wilson Professor of Vision Science
Thesis Supervisor

Accepted by
Leslie A. Kolodziejcki
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Interactive Touch for Manipulation

by

Shaoxiong Wang

Submitted to the Department of Electrical Engineering and Computer Science
on July 15, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Towards helping people in daily life, robots need to better interact with our physical world and inevitably make contact with various objects. Touch provides contact geometry and forces information during interactions, which can be challenging to observe from vision due to occlusions or inherent limitations.

This thesis will focus on how to let robots leverage touch for manipulation in interactive means. We demonstrate several hardware platforms equipped with tactile sensing and integrated perception and control frameworks to apply interactive touch to real-world manipulation tasks. First, we use touch for manipulating deformable objects like cables, using real-time tactile feedback during sliding. The robot can slide and pull the cable into different directions based on the tactile feedback to prevent falling. Second, we perform tactile exploration for learning the physical features of unknown objects. The extracted physical features are further applied to predict the forward model and swing up the in-hand object to a target pose by dynamic motions. Third, we embed tactile sensing with active rollers and design a 6-DoF roller grasper for better in-hand tactile dexterity. We demonstrate that the tactile-enabled roller grasper can robustly perform manipulation tasks for various objects, such as planar object reorientation, rolling along cables with tension, picking and singulating thin objects, etc. We hope applying interactive touch for manipulation can lead us closer to intelligent robot automation and the transformation of our physical world.

Thesis Supervisor: Edward H. Adelson

Title: John and Dorothy Wilson Professor of Vision Science

Acknowledgments

First, I would like to express my deepest gratitude to my advisor, Edward Adelson. Ted has always been my role model. He has infinite wisdom, passion, and persistence. He always tells us deep observations and philosophy effortlessly. His enthusiasm and support make the research experience a great adventure. His patience and attitude towards science guided us to move forward steadily without haste. I believe what I learned from Ted would influence not only my career but also my whole life.

I would also thank my committee members, Alberto Rodriguez, Pulkit Agrawal, and Phillip Isola, for their invaluable suggestions and support during my Ph.D. journey and thesis preparation. Alberto led me into robotic manipulation and has changed my research interests ever since. His insightful suggestions and discussions guided me through this thesis. Pulkit's research and course about sensorimotor and intuitive physics enlightened me a lot during the research. Phil's discussion about the Perceptive Box inspired me to learn physical representation through exploratory interactions. It is my honor to have them on my committee.

I would also like to thank Wenzhen Yuan. She is both a great mentor and friend to me. She taught me everything when I first came to the lab. She shared many valuable suggestions that accompany my Ph.D. journey. Her passion and support encourage me to move forward.

Many other professors generously helped me. I would thank William Freeman, Joshua Tenenbaum, Jiajun Wu, Andrew Owens, Russ Tedrake, Yu She, Kenneth Salisbury, Shan Luo, Daolin Ma, Veronica Santos, and Julie Shah.

I also thank the support and inspiration from researchers of Toyota Research Institute, especially Alex Alspach and Naveen Kuppaswamy.

I would also thank my wonderful collaborators: Siyuan Dong, Branden Romero, Neha Sunil, Chen Wang, Filipe Veiga, Shenli Yuan, Radhen Patel, Megha Tippur, Connor Yako, Achu Wilson, Yuchen Mo, Xingyuan Sun, Roberto Calandra, Michael Lambeta, Po-Wei Chou, Huazhe Xu. It is my pleasure to collaborate with these talented people.

I would also thank my friends for the great times together: Changchen Chen, Nan Du, Tao Du, Xinzhe Fu, Xiaoyue Gong, Beichen Li, Fengyi Li, Jianhua Li, Sandra Liu, Yen-Chen Lin, Yuxiang Ma, Tianyi Peng, Dongying Shen, Yue Wang, Xingyu Wu, Jie Xu, Lei Xu, Youwei Yao, Xiuming Zhang, Zhengdong Zhang, Jialiang Zhao, Yilun Zhou.

Thanks to all the members of the T-Ruth group for the lovely and fun time together.

Special thanks to my roommate and very best friend, Bai Liu, for going on the memorable journey together, through the hills and valleys of life.

Finally, I thank my parents for their unconditional love and support.

Contents

1	Introduction	21
1.1	GelSight sensors for tactile sensing	22
1.2	What is tactile sensing for?	23
1.3	How to use tactile sensing?	25
1.4	Unique Roles of Touch for Manipulation	26
1.5	Contributions	27
2	Cable Manipulation with a Tactile-Reactive Gripper	29
2.1	Introduction	29
2.2	Related Work	32
2.2.1	Contour following	32
2.2.2	Cable/rope manipulation	33
2.2.3	Cloth Manipulation Skills	35
2.2.4	Vision-based Tactile Sensor	36
2.3	Tasks	37
2.4	Method	38
2.4.1	Hardware	38
2.4.2	Perception	43
2.4.3	Control	44
2.4.4	Robotic Cable Manipulation Flowchart	47
2.5	Experiment	47
2.5.1	Experimental Setup	47
2.5.2	Cable following experiments	48

2.5.3	Cable following and insertion experiment	50
2.6	Experimental Results	50
2.6.1	Linear dynamic model evaluation	51
2.6.2	Controller evaluation	52
2.6.3	Generalization to different velocities	54
2.6.4	Generalization to different cables	54
2.6.5	Cable following and insertion	55
2.6.6	Failure cases	56
2.7	Conclusions and Discussion	57
2.7.1	Conclusions	57
2.7.2	Discussions	58
3	SwingBot: Learning Physical Features from In-hand Tactile Exploration for Dynamic Swing-up Manipulation	63
3.1	Introduction	63
3.2	Related Work	66
3.3	Method	68
3.3.1	GelSight	69
3.3.2	Information Fusion for Multiple Exploration Actions	69
3.3.3	Prediction Model for Forward Dynamics	72
3.3.4	Template Objects and Dataset	73
3.4	Experiments	74
3.4.1	Experimental Setup	74
3.4.2	Model Performance	76
3.4.3	Physical Feature Disentanglement	77
3.4.4	Task-oriented Physical Feature	79
3.4.5	Swing-up Results	80
3.5	Discussion and Future Work	80
4	Tactile-Enabled Roller Grasper	83
4.1	Introduction	83

4.2	Related Work	85
4.2.1	Robot Hand for In-Hand Manipulation	85
4.2.2	Vision-based Tactile Sensing	86
4.3	Method	87
4.3.1	Sensor and Hand Design	87
4.3.2	Sensor fabrication	89
4.3.3	Tactile Signal Processing	91
4.3.4	Control methods for In-Hand Manipulation	95
4.4	Results	100
4.4.1	In-Hand Manipulation with Tactile Sensing	100
4.4.2	Efficient object/image reconstruction using Steerable rollers	103
4.4.3	Contributions	104
4.4.4	Limitations and future works	105
5	Conclusion	107
5.1	Summary of Contributions	107
5.2	Future Work	109

List of Figures

1-1	GelSight Wedge. (A) The GelSight Wedge sensor [140] is a compact version of GelSight sensors for robotic gripper, which is used in Chapter 2 and Chapter 3. (B) 3D reconstruction shows the estimated depth of a screw. (C) Yellow arrows show the marker displacement when torsional forces are applied from a screw.	22
1-2	Touch for perception and manipulation. Various tactile tasks can be categorized by their purposes: perception and manipulation. Most tasks lie in the spectrum between perception and manipulation. When the task is more towards manipulation, touch is more used in interactive and dynamic ways.	24
1-3	Passive, active and interactive touch. (A) Passive touch: the action is not determined by the tactile feedback; mainly local information is perceived. (B) Active touch: the action is guided by tactile feedback, without intentionally change the state of the object; global information can be perceived. (C) Interactive touch: the action is reactive based on tactile feedback, with the purpose of changing the state of the object; more manipulation skills can be enabled.	25
1-4	Unique roles of touch for manipulation. Touch provides the contact information under occlusion, and the forceful interaction during manipulation, which is inherently challenging to perceive from vision.	26
2-1	Following a cable with (a) human hands and (b) robotic grippers. . .	30
2-2	The design concept of the cable manipulation system.	39

2-3	Mechanism design. A servo motor drives the slider-crank mechanism via the slider-string-spring system, actuating the parallelogram mechanism via the crank linkage, and finally yielding the motion of opening/closing of the gripper.	40
2-4	Compliant joint design (a) The rigid parallelogram mechanism includes 28 assembly parts. (b) The compliant parallel-guiding mechanism replaces the rigid parallelogram mechanism reducing the assembly parts from 28 pieces to a single piece.	41
2-5	Tactile perception. (a) Gripper with GelSight sensors grasping a cable. (b) Top view of the gripper grasping different cable configurations and the corresponding cable pose estimations. The white ellipse shows the estimation of the contact region. The red and green lines show the first and second principal axes of the contact region, with lengths scaled by their eigenvalues. (c) Top view of pulled cable while the gripper registers marker displacements indicating the magnitude and direction of the frictional forces.	43
2-6	Trade-off between tactile quality and sliding friction. Larger gripping forces lead to higher-quality tactile imprints but difficult sliding. With the same normal force, the grasp quality and friction force varies among cables. The tactile-reactive control adjusts to different cables.	45
2-7	Model cable-gripper dynamics. Schematic diagram of the planar cable pulling modeling.	46
2-8	The robotic cable manipulation flowchart. The flowchart includes three modulus: tactile perception, pose controller, and grip controller.	48
2-9	Experimental setup. UR5 robot arm and two reactive grippers with GelSight sensors.	49

2-10	Cable following experiment. For three instances in time, (a) camera view; (b) pose estimation from tactile imprints, where the yellow line in the center indicates the desired in-hand pose alignment; (c) top view of the trajectory of the end-effector and velocity output of the LQR controller, shown in red. The green dotted line illustrates α . The controller keeps adjusting the cable state in real-time by changing the moving direction to achieve the desired pulling angle.	51
2-11	Predicted vs. actual velocity \dot{y}, $\dot{\theta}$, and $\dot{\alpha}$, of the generalized coordinates of the cable-gripper dynamics, as defined in Fig. 2-7.	52
2-12	Sequence of predicted (orange dash line) and actual (blue solid line) velocity \dot{y}, $\dot{\theta}$, and $\dot{\alpha}$.	53
2-13	Headphone cable following and insertion process. (a)(b) cable following to the plug end, (c) plug on top of the hole with pose mismatch (d) plug pose adjusted and aligning with the hole, (e)(f) cable plugged into the headphone jack on the phone. The plug is labeled with red circle and the headphone jack is labeled with red arrow. . .	55
2-14	Experimental results. Different robot controllers (top), different following velocities (middle), different cables (bottom). For visualization, the three metrics are normalized to $[0, 1]$ by dividing 100%, 0.45 m, and 0.02 m/s respectively (max 1, ideal 1).	56
2-15	Simplified state and dynamics from sliding. The comparison of the global (left) and local (right) perspective of manipulating a piece of cable and fabric. From a global view, it is challenging to model the state and dynamics of a deformable objects due to the large number of degrees of freedom. However, the sliding motion adds constraints to the objects, simplifying the local state and dynamics. It enables fast and reactive manipulation skills.	60

3-1	SwingBot. We develop a learning-based in-hand physical feature exploration method with a GelSight tactile sensor, which assists the robot to perform accurate dynamic swing-up manipulation.	65
3-2	Challenges. Swing-up is a highly-dynamic process, where changing objects' physical properties would have a big impact on the final swing-up angle. Here we show, with the same control parameters, that the dynamics vary when the objects vary: same mass but different center of mass (a)(b); different mass (b)(c); and different friction coefficient (c)(d).	66
3-3	Overview of the architecture. The robot takes several steps to acquire and use the physical features of the held object: (1) <i>Tilting</i> the object at 20° and 45°. The corresponding marker information is encoded by a network with CNN and MLP into a 40-dimensional embedding. (2) <i>Shaking</i> the object. The sequence of marker information is processed by a RNN network into a 40-dimensional embedding. (3) A fusion model concatenates the embedding from both actions and outputs a fused physical feature embedding. (4) A prediction model takes the physical embedding and control parameters as input and outputs a prediction of the final swing-up angle. During training , the whole pipeline is trained in an end-to-end fashion using the final angle for self-supervision. During inference , a set of control parameters are uniformly sampled. The action with the prediction result closest to the goal is selected to perform the swing-up.	68
3-4	Exploration actions and GelSight Signals. The robot executes two in-hand explorations, tilting and shaking, to acquire tactile observations of the object. When tilting, different force and torque distributions are generated by the objects weight can be observed. When shaking, different frictions and vibrations can be observed from temporal sequences of tactile signals.	70

3-5	Template objects. The template objects consist of three components: handle, rack and weights. Different components can be assembled and replaced easily, which creates a variety of objects with different physical properties.	71
3-6	Experiment setup. The GelSight tactile sensor is mounted on a gripper of the robot arm. The recycling system enables automatic data collection.	76
3-7	Task-oriented physical feature visualization: (a) Visualization (with PCA) of the outputted physical embedding (<i>Combined</i>) on the testing samples of the 6 unseen objects (listed in Table. 3.3). (b) Visualization of the data distribution (X-axis: control parameter; Y-axis: final angle) of the testing samples of each object. Each color point refers to one data sample. Objects with similar dynamics are also close to each other on the learned physical embedding space (e.g. 5 and 6). And objects with different dynamics are far away from each other (e.g. 1 and 4).	78
4-1	CAD renderings of the mechanical design of Roller Grasper V4. (A) Fully assembled hand. (B) Optical components required for the GelSight sensor located inside the roller. These components are not rotating with the roller. The non-rotating structure that houses the optical components is called stator. In comparison, the rotating part is called the rotor. (C) The roller (the unbounded rotating mechanism) (D) Arrangement of the camera and mirror inside the roller.	87
4-2	Mold for the seamless roller elastomer. From <i>Left to Right</i> : CAD model of the mold; 3D printed positive mold, with surface smoothed; rubber negative mold; seamless elastomer covering around clear acrylic tube.	89

4-3	Camera calibration and encoder marker. <i>Left:</i> A 6x7 Checkerboard mounted on a calibration tool to get the camera intrinsic matrix, and the corresponding extrinsic matrix in the roller frame; <i>Right:</i> The pattern of the encoder marker to provide precise position encoding, and the corresponding image from the sensor view.	91
4-4	3D reconstruction and marker tracking for the roller sensor. (A) The camera view shows a screw head pressing on the roller sensor, and the 3D shows the estimated 3D reconstruction. (B) The camera view demonstrates the torque exerted on the roller sensor, and the marker displacement visualizes the magnified motion of the markers captured from the sensor. (C) The camera inside the roller sensor captures the raw image, and the sensing area is captured in the mirror. (D) The raw image is unwarped into a rectangular image. (E) The reference image is extracted with the encoder marker from the unwarped image. (F) The difference image is calculated between the unwarped image (after contact) and the reference image (before contact). It is further processed to get the 3D reconstruction, and marker displacement.	93
4-5	In-hand manipulation - cylindrical object.	96
4-6	In-hand manipulation - planar object.	97
4-7	In-hand manipulation - spherical object.	98
4-8	Cable tracing. The roller grasper can reactively roll along cables back and forth without losing the cables. The tactile sensor provides the contact location of the cable and estimated shear forces in real time. The contact location is used to modulate the pivoting angle to compensate for the cable gravity during rolling. The shear force is used to keep the tension during rolling. <i>Left:</i> Without the shear force adjustment, the cable can be accumulate slack over time. <i>Right:</i> With the shear force adjustment, the roller can consistently maintain the tension of the cable over time.	99

4-9	Card picking. We distinguish whether there is only one card picked up by actuating one roller and monitoring the change in shear force. When the number of card within hand is reduced to one, there will be a increase in shear force detected by the tactile sensor.	100
4-10	System Diagram	101
4-11	3D kinematics and roller configurations. (A) Roller Grasper V4 frame definitions. The Two fingers are represented by letters A and B , respectively. Frame O is the hand fixed frame located at the base of the hand. The numerical subscripts represent frames attached at different locations of the hand. Frames 1-5 are attached at different joints while Frame 6 is at the bottom of the roller used as the reference frame for sensor image. The X , Y and Z axes are represented in red, green and blue colors, respectively. Frame O is the world frame with which we reference the manipulation directions. (B) Object rotation in X_O . (C) Object rotation in Z_O or object translation in Y_O , depending on the rolling directions of the two rollers. (D) Object rotation in Y_O or object translation in Z_O , depending on the rolling directions of the two rollers. (Any rotation or translation in directions within $Y_O - Z_O$ plane are possible with different pivot positions) (E) Object translation in X_O . (F) Object screw motion (coupled rotation and translation) .	102
4-12	Roller actuation modes. (A) The roller can hold the current position to grasp the cable, resisting external forces. The marker displacement from the sensor images indicate the exerted external forces. (B) The roller can reactively roll along the cable, following the external forces. (C) The roller can actively roll along the cable, without external forces.	103

4-13 **Surface scanning.** (A) Rolling along a credit card. (B) Stacked tactile images in the time sequence, showing the embossed numbers on the credit card. (C) Processed tactile image with interpolation at the marker region and sharpening filters for better visualization. (D) Rolling along a transparent cup. (E) The embossed characters on the cup. (F) Scanned tactile images stitched in 3D spaces. 103

List of Tables

2.1	Dimension specification.	39
3.1	Quantitative evaluation results of the prediction model with physical embedding from different variants of the fusion model on seen and unseen datasets. The results are shown in degrees.	75
3.2	Quantitative evaluation results of the physical feature disentanglement on both seen and unseen datasets. The metric for the friction is classification success rate (3 classes). The metric for the rest properties is error in percentage (normalized to 0-1 with the minimum and maximum of the value).	77
3.3	Swing-up results on 6 unseen testing objects (with ID 1-6 same as Fig. 3-7). The robot uniformly samples a set of actions and selects the one with the prediction result closest to the final goal to perform the task. In this table, each object is tested 20 trials (5 trials for each desired angle: 45°, 90°, 135° and 180°) and the mean error is listed.	80
4.1	Control Strategies for Manipulation Demonstrations	96

Chapter 1

Introduction

Touch is a natural window for perceiving and interacting with the physical world. When we want to explore unknown objects and manipulate them, many contacts happen between our hands, the objects, and the environment. Based on the contact information from touch feedback, we update the understanding of the objects and make the best decision for the interaction. Since tactile sensing is still at a relatively early stage, previous works on robot touch mainly focus on tactile sensor design and tactile perception [92, 174]. It is non-trivial to build a good tactile sensor and will still require continuous effort to evolve. To demonstrate that the designed tactile sensor is useful, applying it to tactile perception is straightforward and meaningful. However, one most significant advantage of Robotics compared to other artificial intelligence research areas (e.g., Computer Vision, Natural Language Processing, etc.) is the capability to interact with the physical environment, also referred to as Embodied Intelligence [15]. It creates more possibilities when combining tactile sensing with robotic physical interactions. This thesis leverages GelSight tactile sensing[163] for providing rich tactile information. It provides us opportunities to further explore how robots can not only perceive but also interact with the physical world more intelligently through touch.

1.1 GelSight sensors for tactile sensing

This thesis leverage GelSight sensors to give robots a sense of touch. It is a type of camera-based tactile sensor [143, 4, 117, 150, 79, 109, 112]. As a brief introduction, GelSight sensors turn a touch signal into an image. Figure 1-1 shows the signals from GelSight Wedge [140]. It consists of a slab of clear elastomer and reflective skin. When touching objects, the membrane deformation yields a shaded image. By putting a camera and shedding lights from different directions, we can use the photometric stereo to estimate 3D shape. By tracking the motion of the markers on the membrane, we can get observations of shear and torsional force distribution.

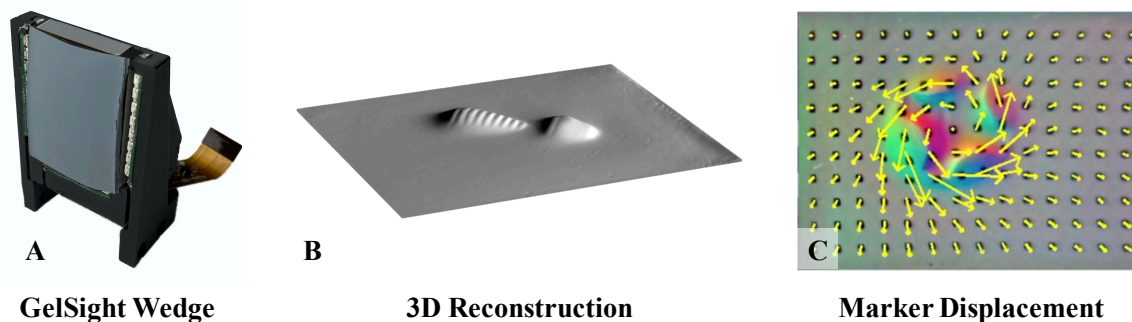


Figure 1-1: **GelSight Wedge.** (A) The GelSight Wedge sensor [140] is a compact version of GelSight sensors for robotic gripper, which is used in Chapter 2 and Chapter 3. (B) 3D reconstruction shows the estimated depth of a screw. (C) Yellow arrows show the marker displacement when torsional forces are applied from a screw.

GelSight sensors significantly lower the threshold for capturing and interpreting touch signals. Cameras are ubiquitous, making the sensor low-cost while providing high-resolution contact geometry and dense normal and tangential force distribution [163, 95]. The raw signals are represented in images or videos, making existing techniques and models in computer vision possible to apply to touch signals. The design principles of the sensor make it flexible to adapt to different shapes for different mechanical benefits, such as slim sensors [38, 140], and round sensors [112, 109, 48]. The softness of the elastomer allows the sensor to comply with different objects, making it suitable for grasping and manipulation.

One thing that distinguishes the GelSight sensor from other camera-based sensors

is getting depth information with minimal effort. The sensor’s illumination is designed for photometric stereo to capture 3D information. Compared with the raw signals, 3D information provides fundamental geometry information, which is a long-pursuing goal of the computer vision community [29, 146, 20]. The 3D information provides a natural way to connect vision and touch in the touch background. The authors in [67] combined a 3D point cloud from vision and touch to track the object’s pose. It is also a rotation-invariant representation, making it suitable for stitching with different poses [88] and efficient learning. The depth can also provide more direct normal forces information [163], and robust contact masks effortlessly, which will be shown in Chapter 2.

With the advantages of GelSight sensors, in this thesis, we try to answer the question of which important tasks would greatly benefit from tactile sensing and how to use tactile sensing better in different settings.

1.2 What is tactile sensing for?

Regarding how robots use touch information, it can be categorized by the purpose. Two significant purposes would be perception and manipulation. When using touch for perception, the goal is to sense and understand the physical world, while when using touch for manipulation, the goal is to change the physical world through interactions, as shown in Figure 1-2.

Extensive research has focused on tactile perception, such as perceiving object material [165, 94], 3D shape [141, 12, 135, 124], class [90, 116, 93], etc. However, we are still in the exploration stage for tactile manipulation since it usually would require integrating sensing, perception, and control to solve the manipulation task. A related direction is grasping. Researchers have explored predicting the grasp stability using tactile feedback [18, 137]. Furthermore, researchers also studied tactile re-grasping policies, which use tactile feedback to guide the next grasp adjustment [16, 56, 21, 76, 30]. However, the primary goal is any stable grasp instead of a targeted grasp for manipulation. It can be viewed as precedent works for tactile sensing for

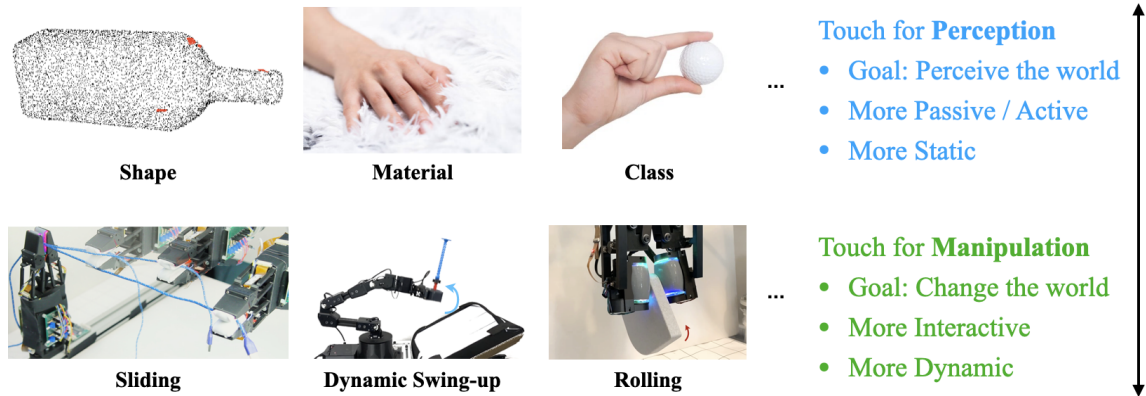


Figure 1-2: **Touch for perception and manipulation.** Various tactile tasks can be categorized by their purposes: perception and manipulation. Most tasks lie in the spectrum between perception and manipulation. When the task is more towards manipulation, touch is more used in interactive and dynamic ways.

manipulation. Some recent works explored applying touch for manipulation in the settings of marble manipulation [131], box manipulation [55], pushing [159], and insertion [34]. This thesis extends this line of work, which is also closely related to Tactile Dexterity [55]. It aims to explore the possible application of touch in various manipulation settings toward improving robot dexterity.

Touch for perception usually focuses more on the static properties of the objects without changing the environment. However, the more touch is used toward manipulation, the more it needs to consider the dynamics of the interaction. For example, researchers have studied using touch for pose tracking [8, 67] or incipient slip [36, 70, 136]. Although these tasks are for perception, they focus more on the dynamic changes of the environment and can be further fed into control for better manipulation.

Most robotic tactile tasks lie in the spectrum between touch for perception and manipulation. They are also often beneficial to each other, such as applying manipulation for better perception (e.g., interactive perception [13]) and applying perception for better manipulation (e.g., perceptive manipulation [77]).

Compared to touch for perception, the benefit of touch can be further exploited by combining it with interaction for manipulation. This thesis will introduce some of the directions that demonstrate the importance and uniqueness of applying touch to

manipulation tasks that are otherwise challenging to solve.

1.3 How to use tactile sensing?

In terms of how to apply touch to robotic tasks, touch could be used passively, actively, or interactively, as shown in Figure 1-3. It has many similar properties analogous to touch for perception and manipulation.

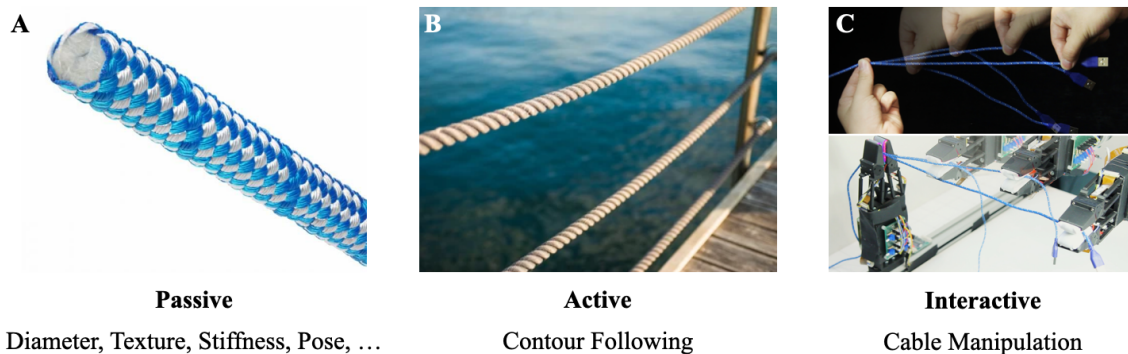


Figure 1-3: **Passive, active and interactive touch.** (A) Passive touch: the action is not determined by the tactile feedback; mainly local information is perceived. (B) Active touch: the action is guided by tactile feedback, without intentionally change the state of the object; global information can be perceived. (C) Interactive touch: the action is reactive based on tactile feedback, with the purpose of changing the state of the object; more manipulation skills can be enabled.

When we apply **passive touch**, we would usually assume that the object’s state is relatively static or will not change much after being touched. We want to perceive some properties based on the tactile signals, such as cloth texture [166, 94], 3D [125], object recognition [90, 116], etc. The action of where to touch is pre-defined and will not adapt to the tactile feedback.

If we go further, when the action of where to touch depends on the tactile feedback, we are moving towards the active or interactive touch. The subtle difference between active and interactive touch is their goals. The major goal of **active touch** is to perceive the world more efficiently and intelligently, such as active contour following [85], active 3D reconstruction[157, 141], active cloth perception [165].

In contrast, the **interactive touch** emphasizes the ability to change the world.

The action not only depends on the tactile feedback but also is applied to interact with and reconfigure the object. Therefore, interactive touch requires us to focus more on the dynamics and control of the system between the robot and objects.

Similarly, there are blurry boundaries between the active and interactive touch, such as tactile SLAM [128], or haptic exploration [108], where we use manipulation as a means to perceive the object’s state. In this thesis, the main focus would be describing the benefits of interactive touch with the purpose of manipulation.

1.4 Unique Roles of Touch for Manipulation



Figure 1-4: **Unique roles of touch for manipulation.** Touch provides the contact information under occlusion, and the forceful interaction during manipulation, which is inherently challenging to perceive from vision.

Compared to vision, touch provides unique information for manipulation, as shown in Figure 1-4. One aspect is to handle occlusion [67, 84]. During manipulation, the hand or gripper inevitably occluded the object. The occlusion makes it challenging to precisely track the object’s pose from vision. For robot manipulation tasks, a slight tracking error can cause severe failure. Touch can provide the contact geometry under the occlusion. With the local contact geometry, the robot can track the object pose or feature more precisely, complementary to global vision.

The other aspect is to provide contact force information during the interaction [34,

84]. It is inherently difficult to observe forces from vision. However, touch provides direct force information between the hand, the manipulated object, and the environment. The robot can use the force information to make better decisions, such as modulating the minimal grasping force, keeping the desired force, adjusting poses, switching motion primitives, etc.

This thesis will explore how to use the contact geometry and forces during interactions towards dexterous manipulation in different settings.

1.5 Contributions

This thesis aims to discover the importance of touch for robotic tasks. We propose to apply robot touch for manipulation in more interactive ways. Compared to the previous focus on tactile perception [92, 163], the application to manipulation tasks emphasizes the dynamic interactions between robots and objects. Touch provides unique contact geometry and forces that are challenging to capture from vision. Applying touch for manipulation requires more system integration between sensing, perception, control, and planning, but eventually can lead us closer to robot automation for transforming our physical world. We will introduce the detailed methods to apply interactive touch for manipulation to increase robot dexterity in different settings.

In Chapter 2, we introduce a motion primitive of sliding enabled by touch for cable manipulation. We use real-time tactile feedback control to follow a dangling cable to the end. The robot can perceive the real-time pose of the cable and the friction forces during the cable sliding. It interactively perceives and changes the cable states by pulling the cable in different directions. We decouple the control into cable grip control and cable pose control. The tactile-reactive behavior turns a complex task of manipulating a highly deformable object with uncontrolled variations in friction and shape into an achievable task.

In Chapter 3, we explore applying touch for dynamic manipulation tasks, where we can increase robot dexterity through dynamic motion. The goal is to swing up

the in-hand object to a target pose by dynamic motions. Since the task is sensitive to the object’s physical properties, we propose tactile exploration, which provides touch signals under different exploration interactions. The exploration data is later used to extract physical features of the object and predict the forward model for control. The tactile exploration with the self-supervised learning framework greatly improves the performance of the dynamic manipulation task.

In Chapter 4, we explore improving in-hand dexterity by combining the rolling motion for manipulation with tactile feedback. We design a 6-DoF roller grasper equipped with vision-based tactile sensors and the perception algorithms to handle the rolling contact. We demonstrate that the grasper can adjust its joints based on the tactile signals to manipulate various objects continuously, such as planar object reorientation, rolling along cables with tension, card singulation, etc. Combining tactile sensors with dexterous hands opens up a whole new range of possibilities for robot in-hand manipulation.

We conclude and discuss future directions in Chapter 5.

Chapter 2

Cable Manipulation with a Tactile-Reactive Gripper

We start our journey by exploring deformable linear objects manipulation, such as cables, with tactile-reactive control. The real-time tactile signals enable the motion primitive of sliding. Compared to the previous works of tactile contour following with fixed objects [99, 85], we propose to incorporate cable-gripper dynamics and control, and tactile regrasping policy to manipulate cables in the free space. The tactile sliding also makes it more efficient and requires fewer constraints compared to previous tabletop manipulation for deformable linear objects with vision [107, 155, 127].

2.1 Introduction

Contour following is a dexterous skill which can be guided by tactile servoing. A common type of contour following occurs with deformable linear objects, such as cables. After grasping a cable loosely between the thumb and forefinger, one can slide the fingers to a target position as a robust strategy to regrasp it. For example, when trying to find the plug-end of a loose headphone cable, one may slide along the cable until the plug is felt between the fingers.

Cable following is challenging because the cable’s shape changes dynamically with the sliding motion, and there are unpredictable factors such as kinks, variable friction,

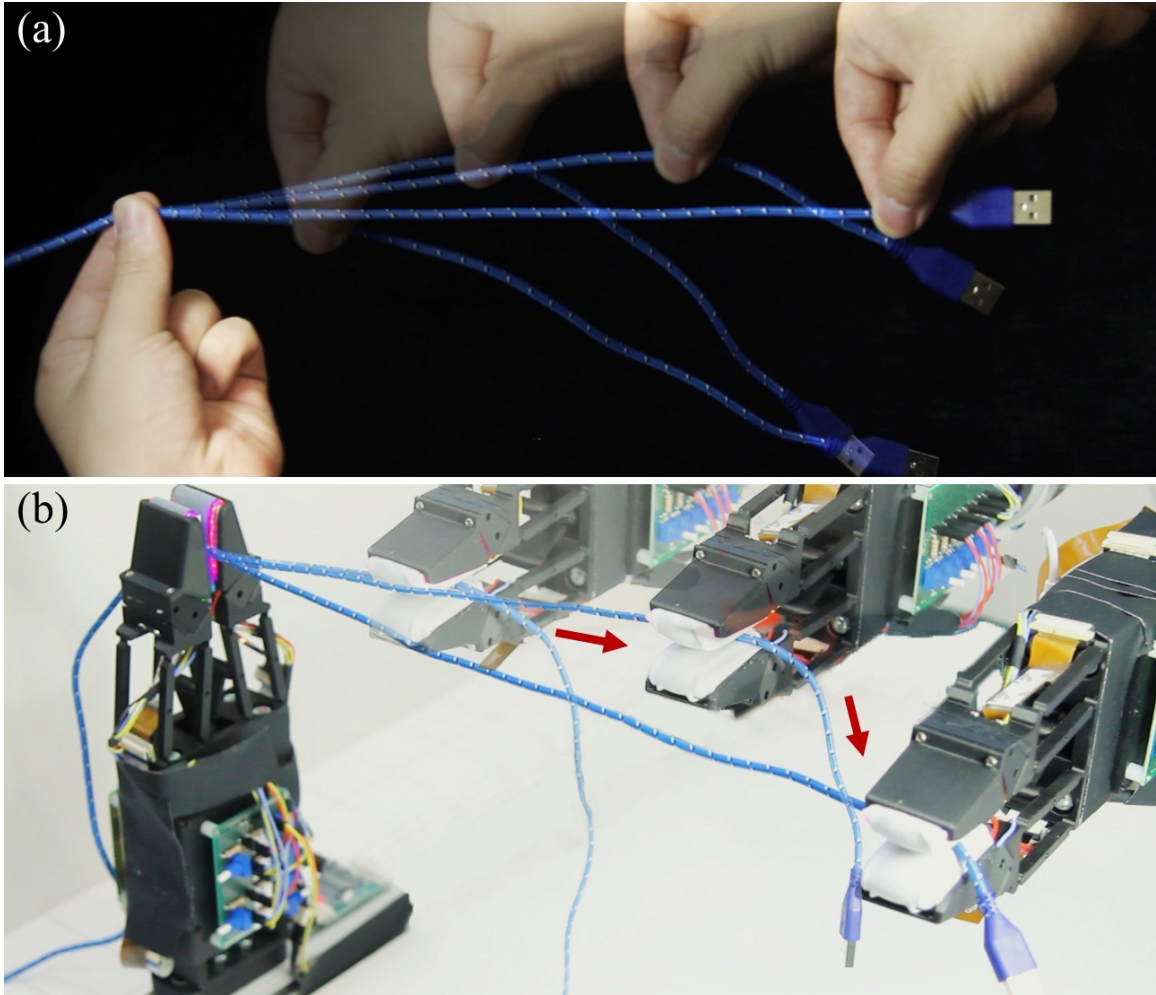


Figure 2-1: Following a cable with (a) human hands and (b) robotic grippers.

and external forces. For this reason, much work on cables (and other deformable linear objects) has utilized mechanical constraints [155, 172, 107]. For example a rope may be placed on a table, so that gravity and friction yield a quasistatic configuration of the cable. A gripper can then adjust the rope configuration, step by step, at a chosen pace.

Our goal is to manipulate cables in real time, using a pair of grippers, with no added mechanical constraints. The cables are free to wiggle, swing, or twist, and our grippers must rapidly react using tactile feedback. In particular, we look at the task of picking one end of a cable with a gripper and following it to the other end with a second gripper, as shown in Fig. 2-1.

We designed a novel gripper that is lightweight and fast reacting, and equipped

it with high resolution tactile sensors. This novel hardware, when paired with appropriate control policies, allows us to perform real-time cable following in free space.

In this paper we do not use vision, relying on tactile sensing alone. While vision can be helpful, we are able to perform the task purely with tactile guidance. Deformable linear objects are easily occluded from view by grippers, by the environment, and often by itself. Tactile perception allows for precise localization once the cable is grasped. Tactile active perception, like when pulling from the two ends of a cable until it is in tension, can also be used to simplify perception such as in the case of a tangled rope.

We approach cable following by dividing the desired behavior into two goals: 1) Cable grip control, which monitors the gripping force to maintain friction forces within a useful range; and 2) Cable pose control, which regulates the configuration of the cable to be centered and aligned with the fingers. These two controllers work in tandem to enable smooth and efficient cable sliding. The first controller maintains the frictional interaction between cable and fingers near a desired working point, which simplifies the dynamics that the cable pose controller has to regulate. To accomplish this task, we build a system with the following modules:

- **Tactile-reactive gripper.** We design a parallel-jaw gripper with force and position control capabilities (Sec. 2.4.1), fitted with GelSight-based tactile sensors [162] yielding a 60Hz grip bandwidth control.
- **Tactile perception.** We estimate in real-time the pose of the cable in the gripper, the friction force pulling from the gripper, and the quality of the tactile imprints (Sec. 2.4.2).
- **Cable grip controller.** The gripper regulates the gripping force by combining a PD controller and a leaky integrator that modulates the friction force on the cable, and provides grip forces to yield tactile imprints of sufficient quality for perception (Sec. 2.4.3).
- **Cable pose controller.** The robot controls the cable configuration on the

gripper fingers with an LQR controller, based on a learned linear model of the dynamics of the sliding cable (Sec. 2.4.3).

We evaluate the complete system in the task of cable following for various cables, sliding at different velocities, and benchmark against several baseline algorithms. The results in Sec. 2.6 show that training the system on a single cable type allows generalization to a range of cables with other physical parameters. Finally, we demonstrate a robot picking a headphone cable, sliding the fingers until feeling the jack connector, and inserting it, illustrating the potential role of the system in complex active perception and manipulation tasks. Video demonstrations of the aforementioned tasks are available on the project website at <http://gelsight.csail.mit.edu/cable/>.

2.2 Related Work

In this section we review work relevant to contour following and cable manipulation.

2.2.1 Contour following

Contour following of rigid objects has been widely studied using both visual [81] and tactile perception [22, 144]. These techniques do not directly translate to deformable objects due to dynamic shape changes that are difficult to model, especially in real time.

The most similar contour following work to ours is by [53], who proposed a reinforcement learning approach to close a deformable ziplock bag with feedback from BioTac sensors. The work demonstrated a robot grasping and following the edge of the bag. In contrast to our approach, they use a constant grasping force and discrete slow actions. As a consequence, they achieve a maximum speed of 0.5 cm/s, compared to 6.5 cm/s in our work.

2.2.2 Cable/rope manipulation

Manipulating deformable linear objects (DLOs) has attracted attention in the robotics community [58] with tasks including tying knots [105, 113], untangling [91, 50], insertion [142], reshaping [155, 172, 107], surgical suturing [101], or dynamic rope manipulation [152, 171]. Our approach to cable manipulation through tactile perception and control is fundamentally distinct from the existing literature and enables a larger potential action space.

Approaches

Much of the classical work in DLO manipulation involves perceiving the state of the DLO, simulating the dynamics of the DLO, or planning its motion. Visual perception for DLO state estimation is difficult given the infinite dimensional configuration space with the object’s shape dynamically changing while often being occluded. [105] described rope state topologically by listing intersections created by rope crossings. Methods that more completely describe location along the entire length of the DLO often use non-rigid registration techniques to track the rope from a known initial state [130, 24]. Alternatively, an initial state estimate from a given point cloud can be refined to better align with the system dynamics [91, 71].

The most common methods for simulating DLOs use mass-spring models [139], energy minimization [10], or finite-element methods (FEM) [110]. These models can be computationally expensive and require knowledge of the DLO’s physical properties such as rigidity, elasticity, and friction. [9] avoids an explicit deformable object model by using an approximation to the Jacobian of the deformable object to drive object points to a target set. Work by [153] also avoids complicated dynamics models by moving the cables at high enough speeds that they assume each rope segment follows the motion of the robot with a constant time delay.

Motion planning for DLOs has traditionally used sampling-based approaches such as a probabilistic roadmap (PRM) [75] or Rapidly-exploring Random Trees RRTs

[82]. [104] used these methods to create local planners based on minimum energy curves. For knot-tying, [113] plan long-horizon, complex motions by simulating deformations of a rope in response to random external forces and placing configurations that would be part of the knot’s topological forming sequence in a PRM.

Learning-based approaches can help simplify aspects of the problem. Given the inherent difficulty of complete state estimation, some DLO manipulation works are trained directly from visual data without explicitly estimating the full state [51]. [107] learns a pixel-level inverse dynamics model for a rope with self-supervised autonomous pick and place interactions. More recently, [127, 47] use dense object descriptors to find pixel-wise correlations between images of ropes trained in simulation. Another class of work learns dynamics models for rope and uses them with Model Predictive Control (MPC). Work by [39] learns a video prediction model. While these methods work for short-horizon tasks like shaping, planning for more complex, long-horizon tasks like knotting requires more guidance, for example learning from demonstration. While these data-driven methods allow for faster computation, they are less generalizable for other DLO manipulation tasks.

Rope Manipulation Skills

Due to their high dimensional dynamics, manipulating deformable linear objects is usually simplified by constraining their motion with external features, for example against a table [155, 172, 107], with additional grippers [101], or pegs [113]. Another common strategy involves limiting movements to long series of small deformations with pick and place actions [155, 107]. Thus, the dynamics of the system can be treated as quasistatic.

Furthermore, the action space in DLO manipulation literature is generally limited to those using fixed grasps of the DLO. Besides pick and place, other actions include following specific, potentially dynamic, trajectories [152], moving a segment of a rope using two grippers [103], insertion [142], and wrapping [173].

Few works exploit sliding along the DLO. [173] routed a cable around pegs using one end-effector that was attached to the cable end and another fixed end-effector

that would passively let a cable slip through in order to pull out a longer length of cable. This system, while allowing for sliding, loses the ability to sense and control the state of the cable at the sliding end, which can be in contact with any point of the cable.

[73] traces cables in a wire harness using a gripper with rollers in the jaws. This gripper passively adjusts grip force using springs to accommodate different sized cables. They sense and control the force perpendicular to the translational motion along the cable in order to follow the cable. The cables in our work are considerably smaller and less rigid, so such forces would be difficult to sense.

Furthermore, both of the specialized, passive end-effectors in the above two works have limited capabilities beyond sliding along the cable. In our work, the parallel jaw gripper used to follow a cable is also used to insert the cable into a headphone jack, demonstrating the potential of this hardware setup for additional tasks.

Another example of work involving sliding with rope from [154] shows how our framework could potentially be extended for the knot-tying task. To pass one end of the rope through a loop, they leverage tactile sensing to roll the two rope ends relative to each other in between the fingers.

2.2.3 Cloth Manipulation Skills

A further suggestion of generality of the approach is the potential use of sliding in fabric manipulation. Most work in fabric manipulation similarly uses the quasistatic assumption and incremental pick and place movements [148, 61, 46]. However, the sliding skill simplifies the task of finding two adjacent corners in order to fold a piece of fabric. [114] holds up a corner of the fabric and uses gravity to trace straight down to find the second corner without sensory feedback. Similarly, [167] executes a “pinch and slide” motion along the top edge of a piece of fabric.

Besides the knotting example from [154], force and tactile sensors can be seen in a variety of rope manipulation literature. [1] uses a force torque sensor to detect changes in contact state, for example the rope moving from free space to contacting

an edge of a rigid object. [168] detects vibration frequency of a rope using a force torque sensor before counteracting the vibration.

2.2.4 Vision-based Tactile Sensor

The vision-based tactile sensor converts touch to vision by using a camera to visualize the deformation of the contact surface. With its high spatial resolution, this type of sensor shows unique advantages and has been successfully utilized in different robotic manipulation tasks, for instance, contour following [86], cutting [151], dish loading [78], in-hand manipulation [80], etc.

As a popular vision-based tactile sensor, the GelSight sensor [162] can recover a precise depth map of the contact surface with designed three light illumination. The measured high resolution local contact geometry can be used to estimate the object state. Here we use it to estimate the pose of the cable in hand. Fig. 2-2 shows an example of the sensor raw output and recovered depth image when grasping a cable. The sensor also measures approximate shear force by tracking the black markers on the sensor surface [164]. Here we use it to estimate the approximated friction force during cable sliding.

The GelSight sensor has been extensively applied in various manipulation problems. [89] implemented a USB insertion task from random grasping poses based on object pose estimation feedback from the sensor. [68] used the 3D point cloud from GelSight sensor in a Kalman filter to better register the position of a screwdriver in a peg-in-hole task. [17] and [57] used the tactile images to evaluate the quality of a grasp and further infer better regrasp positions. [35] used the sensor to predict slip and used its slip signal to modulate grasping forces while conducting a bottle cap screwing task. [132] proposed a tactile-based model predictive control method to reposition an object. [37] trained a tactile-based object insertion policy that could correct small misalignment between the object and the environment. [55] designed closed-loop tactile controllers for dexterous table-top manipulation with dual-arm robotic palms, where similar idea to our method of simultaneously controlling contact state

and object state was adopted. [138] implemented a task of swing an elongated object to a target pose based on the learned friction, center of mass properties of the grasp object with the GelSight sensor.

2.3 Tasks

Cable Following The goal of the cable following task is to use a robot gripper to grip the beginning of the cable with proper force and then control the gripper to follow the cable contour all the way to its tail end. The beginning end of the cable is initially firmly gripped by another fixed gripper during the cable following process. The moving gripper is allowed to regrasp the cable by bringing it back to the fixed gripper, resulting in two-hand coordination with one of the hands fixed. Several cables with different properties (shape, stiffness, surface roughness) are tested here for generalization.

Cable Insertion The goal of the cable insertion task is to find the plug at the end of a cable and insert the plug into the socket. We show that this can be done by leveraging the ability to slide the fingers on the cable, and demonstrate it with a headphone cable with a cylindrical jack connector at its end.

Robot System In order to tackle this task, the following four hardware elements are necessary:

- tactile sensor to measure the grasped cable position and orientation in real time
- tactile sensor to measure the amount of friction force during sliding in real time
- fast reactive gripper to modulate the grasping force according to the measured friction
- fast reactive robot arm to follow the measured cable orientation and keep the measured cable position in the center of the gripper.

2.4 Method

The key idea of our method is to use tactile control to monitor and manipulate the position and forces on the cable between the fingers. The concept is illustrated in Fig. 2-2. We divide the tactile controller into two parts:

- i. **Cable Grip Control** so the cable alternates between firmly grasped and sliding smoothly,
- ii. **Cable Pose Control** so the cable remains centered and aligned with the fingers when pulling and sliding.

This decomposition can be seen as an application of the tactile dexterity framework in [55] applied to a sliding primitive and to deformable object manipulation. In this section, we describe the implementation of the tactile controller by introducing the reactive gripper, the tactile perception system, the modeling of the cable, and the two controllers.

2.4.1 Hardware

Most commercialized robotic grippers do not offer sufficient bandwidth and low latency for real-time feedback control. To that end we design a parallel gripper with 2 fingers (with a revised GelSight sensor), a compliant parallel-guiding mechanisms, and slide-crank linkages actuated by a servo motor as shown in Fig. 2-3.

Mechanism design Parallelogram mechanisms are widely used to yield lateral displacement and slider-crank mechanisms are broadly employed to actuate the parallelogram mechanism for parallel grippers. We use them to facilitate parallel grasping. To make a compact actuation mechanism, we use a tendon-driven system.

One end of a string (tendon) is tied to a motor disk which is fixed on the servo motor installed in a motor case. The other end of the string is tied to the slider as shown in Fig. 2-3. We use a compression spring between the slider and the motor box with pre-tension forming a slider-string-spring system. The string then passes through a pulley to change its direction. One end of the crank linkage is connected to

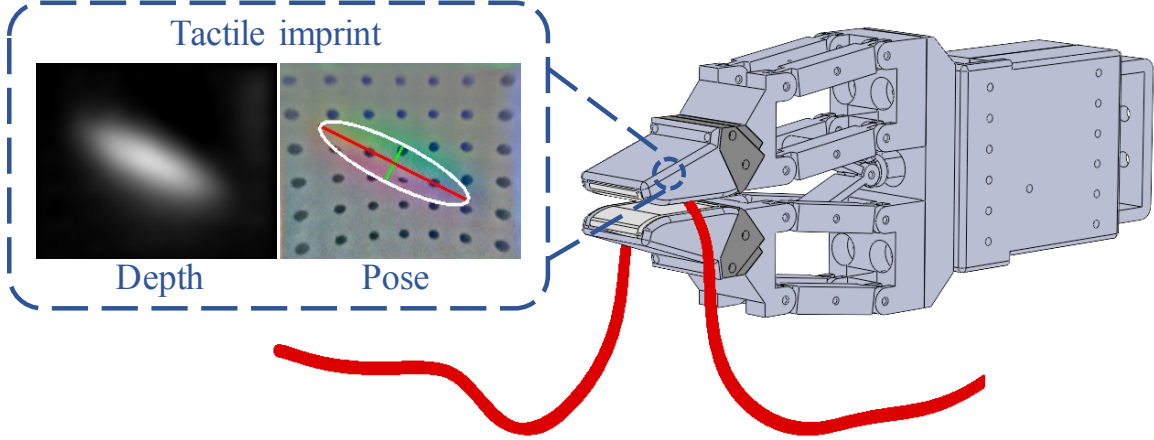


Figure 2-2: The design concept of the cable manipulation system.

Table 2.1: Dimension specification.

Parameters	Value	Parameters	Value
r_1	15 mm	l_1	23.25 mm
r_2	50 mm	l_2	100 mm
r_3	30 mm	θ_2^i	127°
r_4	64 mm	θ_3^i	307°

the slider and the other is coupled with the rocker of the parallelogram mechanism. The finger is attached to the coupler of the parallelogram mechanism. The string drives the slider down, actuating the parallelogram mechanism via the crank linkage and producing the desired lateral displacement of the finger. Two fingers assembled symmetrically around the slider yields a parallel gripper.

Mechanism dimensions The next step is to determine the dimensions of the gripper. The design guidelines are as follows: 1) The max opening of the gripper is targeted at 100 mm, i.e., 50 mm displacement for each finger; 2) The parallelogram mechanism should fit the size of the revised GelSight fingertips; 3) Reduce overall size and weight of the gripper as much as possible. According to the kinematics of the parallelogram and slider-crank mechanism as well as the aforementioned constraints, we designed a gripper with the dimensions in Table 2.1. Refer to Fig. 2-3 for the definition of all variables. Note that θ_2^i and θ_3^i are the initial values of θ_2 and θ_3 .

Compliant joint design The original gripper design is comprised of four sets of rigid parallelogram mechanisms (two sets on each side), which contains 28 assembly

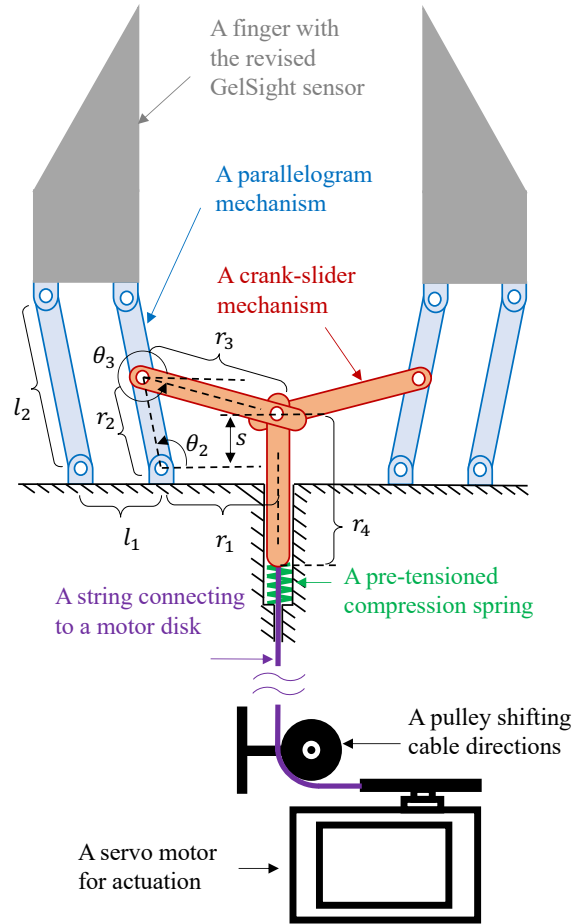


Figure 2-3: **Mechanism design.** A servo motor drives the slider-crank mechanism via the slider-string-spring system, actuating the parallelogram mechanism via the crank linkage, and finally yielding the motion of opening/closing of the gripper.

pieces and consumes assembly time. Compliant mechanisms [63] can produce exactly the same motion as those produced by rigid body mechanisms, but greatly reduce the part count and assembly procedures. We consider to use compliant joints to simplify the parallelogram mechanism.

Modeling and analysis of compliant mechanisms is however more complex than that of rigid-body mechanisms due to their infinite degrees of freedom (DOFs) and complex deformations. Screw theory-based methods [59, 60, 106], beam theory [134, 6], topological synthesis [45], and Pseudo-Rigid-Body (PRB) model [65, 66] are the common methods to model and analyze the compliant mechanisms. Among those approaches, the PRB model bridges the compliant mechanisms and rigid body theories,

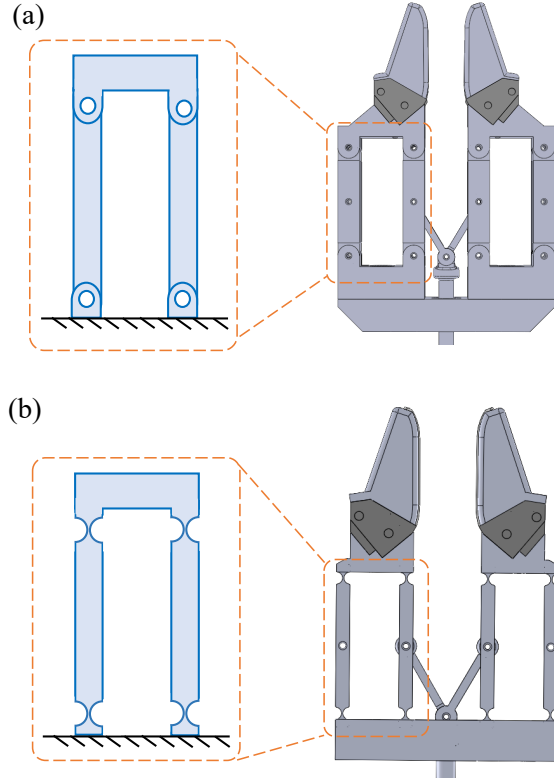


Figure 2-4: **Compliant joint design** (a) The rigid parallelogram mechanism includes 28 assembly parts. (b) The compliant parallel-guiding mechanism replaces the rigid parallelogram mechanism reducing the assembly parts from 28 pieces to a single piece.

which will be used in this work.

The PRB model provides a simple and computationally efficient solution to analyze kinematics and statics (kinetostatic) of the compliant mechanisms. Given a specific compliant mechanism, the kinetostatic analysis of the mechanism with the PRB model is referred to the forward analysis. If we know a particular rigid body mechanism, the design of a corresponding compliant mechanism is referred to the inverse analysis. In this work, we consider to leverage the latter one to design a corresponding compliant mechanism to replace the rigid parallelogram mechanisms.

The rigid parallelogram mechanism and the corresponding gripper are as shown in Fig. 2-4a. Considering the revolute joint as the pseudo rigid joint, one can replace it by the living hinge with the inverse analysis. The living hinge is a special form of a flexural pivot with little resistance throughout its deflection [64]. With the sub-

stituted living hinges, one can convert the mechanisms in Fig. 2-4a to the compliant parallel-guiding mechanism and the corresponding gripper in Fig. 2-4b.

We use the living hinge design to convert the rigid parallelogram mechanism in Fig. 2-4a to an equivalent compliant parallel-guiding mechanism to reduce the assembly process. The living hinge design reduces the 28 pieces of the rigid mechanism to a single part while offers the approximately same kinematics functionality. The overall size of the final prototype has length 260 mm, width 140 mm, and thickness 85 mm at the rest position.

Actuation We select a high torque and high speed servo motor, dynamixel XM430-W210-T from Robotis, as the actuator for the gripper. It offers 77 rpm no-load speed and 3 N.m stall torque. According to the kinematics analysis of the crank-slide mechanism, we map the motor speed (Ω) to the gripping speed (V) as:

$$\begin{cases} V = 2\dot{\theta}_2 l_2 \\ \dot{\theta}_2 = \dot{s} \frac{\cos \theta_3}{r_2 \sin(\theta_3 - \theta_2)} \\ \dot{s} = \Omega r_d, \end{cases} \quad (2.1)$$

where r_d is the radius of the motor disk, and s is the displacement of the slider. Similarly, we map the motor torque (τ) to the gripping force (F) according to the energy method and free body diagram (FBD):

$$\begin{cases} F = \frac{M_2}{l_2 \sin \theta_2} \\ M_2 = \frac{P \dot{s}}{\dot{\theta}_2} = \frac{P r_2 \sin(\theta_3 - \theta_2)}{\cos \theta_3} \\ P = \frac{\tau}{r_d} - k s, \end{cases} \quad (2.2)$$

where M_2 is the reaction torque at θ_2 given a grip force F at the fingertip, P is the reaction force at the slide in the vertical direction corresponding to M_2 , and k is the stiffness of the compression spring.

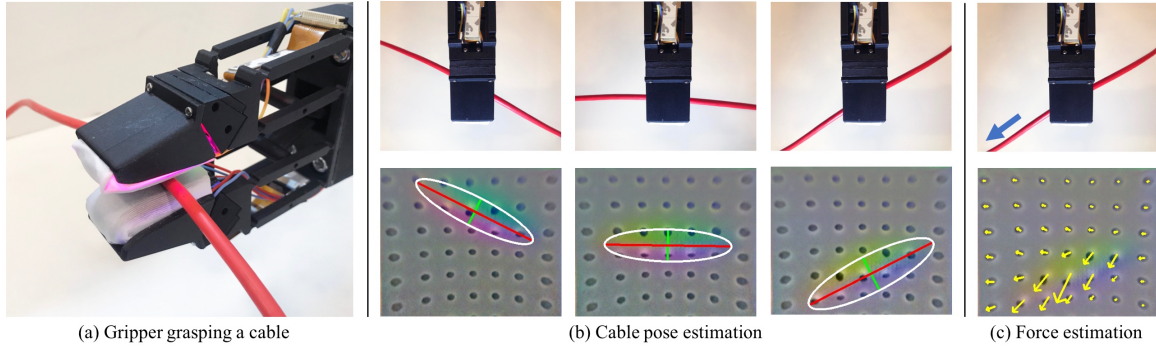


Figure 2-5: **Tactile perception.** (a) Gripper with GelSight sensors grasping a cable. (b) Top view of the gripper grasping different cable configurations and the corresponding cable pose estimations. The white ellipse shows the estimation of the contact region. The red and green lines show the first and second principal axes of the contact region, with lengths scaled by their eigenvalues. (c) Top view of pulled cable while the gripper registers marker displacements indicating the magnitude and direction of the frictional forces.

2.4.2 Perception

Figure 2-5 illustrates the process to extract cable pose, cable force and grasp quality from tactile images.

Cable pose estimation First, we compute depth images from the raw tactile images by estimating surface normal and applying Fast Poisson Solver (FPS) for integration [162]. Then, we extract the contact region by thresholding the depth image. Finally, we use Principal Component Analysis (PCA) on the contact region to get the principal axis of the imprint of the cable on the sensor.

Cable friction force estimation We use blob detection to locate the center of the black markers in the tactile images [35]. Then we use a matching algorithm to associate marker positions between frames, with a regularization term to maintain the smoothness of the marker displacement flow. We compute the mean of the marker displacement field (D), which is approximately proportional to the resultant friction force.

Cable grasp quality In this task, we evaluate the grasp quality (S) based on whether the area of the contact region is larger than a certain area. A tactile imprint with poor quality (small contact region) will give noisy and uncertain pose estimation. By increasing the grasping force, as shown in Fig. 2-6, we can increase the grasp quality.

2.4.3 Control

Cable Grip Controller The goal of the grip controller is to modulate the grasping force such that 1) the friction force stays within a reasonable value for cable sliding (too small and the cable falls from the grip, too large and the cable gets stuck), and 2) the tactile signal quality is maintained. We employ a combination of a PD controller and a leaky integrator. The PD controller uses the mean value of the marker displacement (D) (approximately proportional to the resultant frictional force) as feedback and regulates it to a predefined target value (D_t). We use position control to modulate gripping force with the following PD controller for the reference position u_{pd} of the servo motor:

$$\begin{aligned} u_{pd}[n] &= K_p e[n] + K_d (e[n] - e[n-1]) \\ e[n] &= D[n] - D_t[n], \end{aligned} \tag{2.3}$$

where K_p and K_d are the coefficients for the proportional and derivative terms, and $D[n]$ is the measured mean value of the marker displacement. The leaky integrator raises D_t of the PD controller if the signal quality (S) is poor as follows:

$$\begin{aligned} D_t[n] &= \alpha D_t[n-1] + (1-\alpha)(1-S), \\ S &= \begin{cases} 1 & \text{if good quality} \\ 0 & \text{if poor quality} \end{cases} \end{aligned} \tag{2.4}$$

where α is the leakage at each time step and S is the signal quality indicator. If $S = 1$, the desired reference friction D_t leaks. If $S = 0$, the desired reference friction D_t increases.

Cable-Gripper Dynamics Model We model the cable-gripper dynamics as a planar pulling problem. As shown in Fig. 2-7, the region of the cable in contact with the tactile sensor (blue rectangle on the right) is represented as a 2D rigid sliding object. We parameterize its position and orientation with respect to X axis of the sensor frame with y and θ . We further define the angle α between the center of the cable

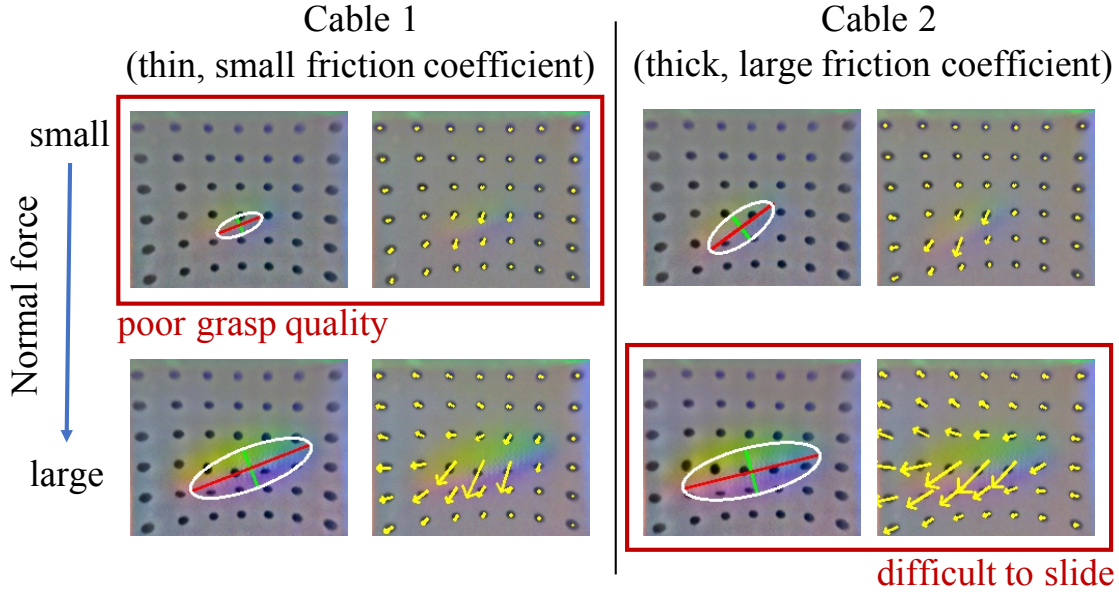


Figure 2-6: **Trade-off between tactile quality and sliding friction.** Larger gripping forces lead to higher-quality tactile imprints but difficult sliding. With the same normal force, the grasp quality and friction force varies among cables. The tactile-reactive control adjusts to different cables.

on the moving gripper and the orientation of the fixed gripper (blue rectangle on the left). These three parameters $[y \ \theta \ \alpha]$ define the state of the cable-gripper system. We finally define the control input on the system ϕ as the pulling direction relative to the angle α (labeled with red arrow).

Since a deformable gel surface has complex friction dynamics, we use a data-driven method to fit a linear dynamic model rather than first principles. The state of the model is $\mathbf{x} = [y \ \theta \ \alpha]^T$, the control input $\mathbf{u} = [\phi]$, and the linear dynamic model:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \tag{2.5}$$

where A and B are the linear coefficients of the model.

To efficiently collect data, we use a simple proportional (P) pulling controller as the base controller supplemented with uniform noise for the data collection process. The P controller controls the velocity of the robot TCP in the y axis and we leave the velocity in the x axis constant. The controller is expressed in Equation 2.6, where K_p^v

is the coefficient of the proportional term, and $N[n]$ is random noise sampled from a uniform distribution $[-0.01, 0.01]$. The intuition for this baseline controller is that when the robot (sensor) moves to the $+\vec{y}$ direction in world frame and composes a $+\alpha$ angle, the cable gets pulled from the opposite direction $-\vec{y}$ and dragged back to the center of the gripper if it is initially located in the $+\vec{y}$ region. Similar idea applies to the opposite scenarios.

$$v_y[n] = K_p^v y[n] + N[n] \quad (2.6)$$

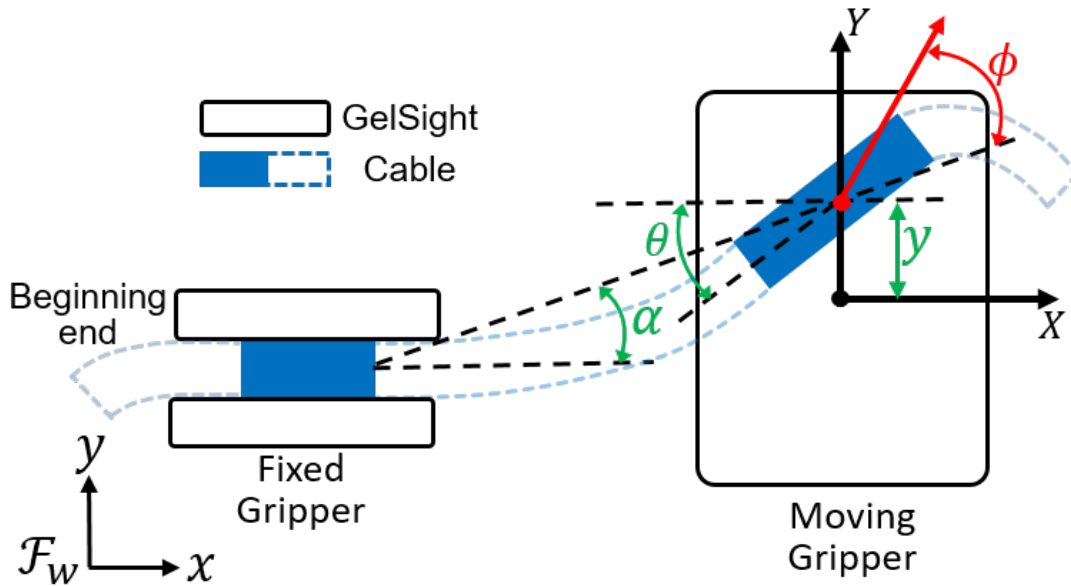


Figure 2-7: **Model cable-gripper dynamics.** Schematic diagram of the planar cable pulling modeling.

Cable Pose Controller The goal of the cable pose controller is to maintain the cable position in the center of the tactile sensor ($y^* = 0$), the orientation of the cable to be parallel to the X axis ($\theta^* = 0$) and the inclination of the pulled cable also parallel to the X axis ($\alpha^* = 0$). The nominal trajectory of the cable pose controller $(\mathbf{x}[\mathbf{n}]^*, \mathbf{u}[\mathbf{n}]^*)$ is then constant and equal to $([0 \ 0 \ 0]^T, [0])$, that is, regulating around zero.

We formulate an LQR controller with the A and B matrices from the linear model and solve for the optimal feedback gain K , which in turn gives us the optimal control

input $\bar{\mathbf{u}}[n] = -\mathbf{K}\mathbf{x}[n]$, where $\mathbf{x} = [y \ \theta \ \alpha]^T$ as shown in Fig 2-7. The parameters of the LQR controller we use are $\mathbf{Q} = [1, 1, 0.1]$ and $\mathbf{R} = [0.1]$, since regulating y and θ (making sure the cable does not fall) is more important than regulating α (maintain the cable straight).

2.4.4 Robotic Cable Manipulation Flowchart

An overview of the robotic cable manipulation flowchart is given in Fig 2-8, which includes the major components of the system: tactile perception, pose controller, and grip controller. The flowchart presents an overview of the perception and control algorithms. The perception module takes the raw feedback (raw image) from the tactile sensors as the input, and generates the cable pose, marker displacement, and contact area as the outputs. On one hand, the cable pose is fed to the pose controller, generating action commands applied to the robot to modulate the pose of the cable in the gripper. On the other hand, the marker displacement and contact area are fed to the grip controller, generating action commands applied to the gripper to modulate the gripping force. Note that the flowchart reflects the logic of the cable following task, and the insertion task is not included in the flowchart for the sake of brevity.

2.5 Experiment

2.5.1 Experimental Setup

The experimental setup in Fig. 2-9 includes a 6-DOF UR5 robot arm, two reactive parallel-jaw grippers (as described in Section 2.4.1) and two pairs of revised fingertip GelSight sensors attached to the gripper fingers. One of the grippers is fixed on the table and another one is attached to the robot. The control loop frequencies of the UR5 and the gripper are 125 Hz and 60 Hz, respectively.

We use five different cables/ropes to test the controllers (Fig. 2-14 bottom): Thin

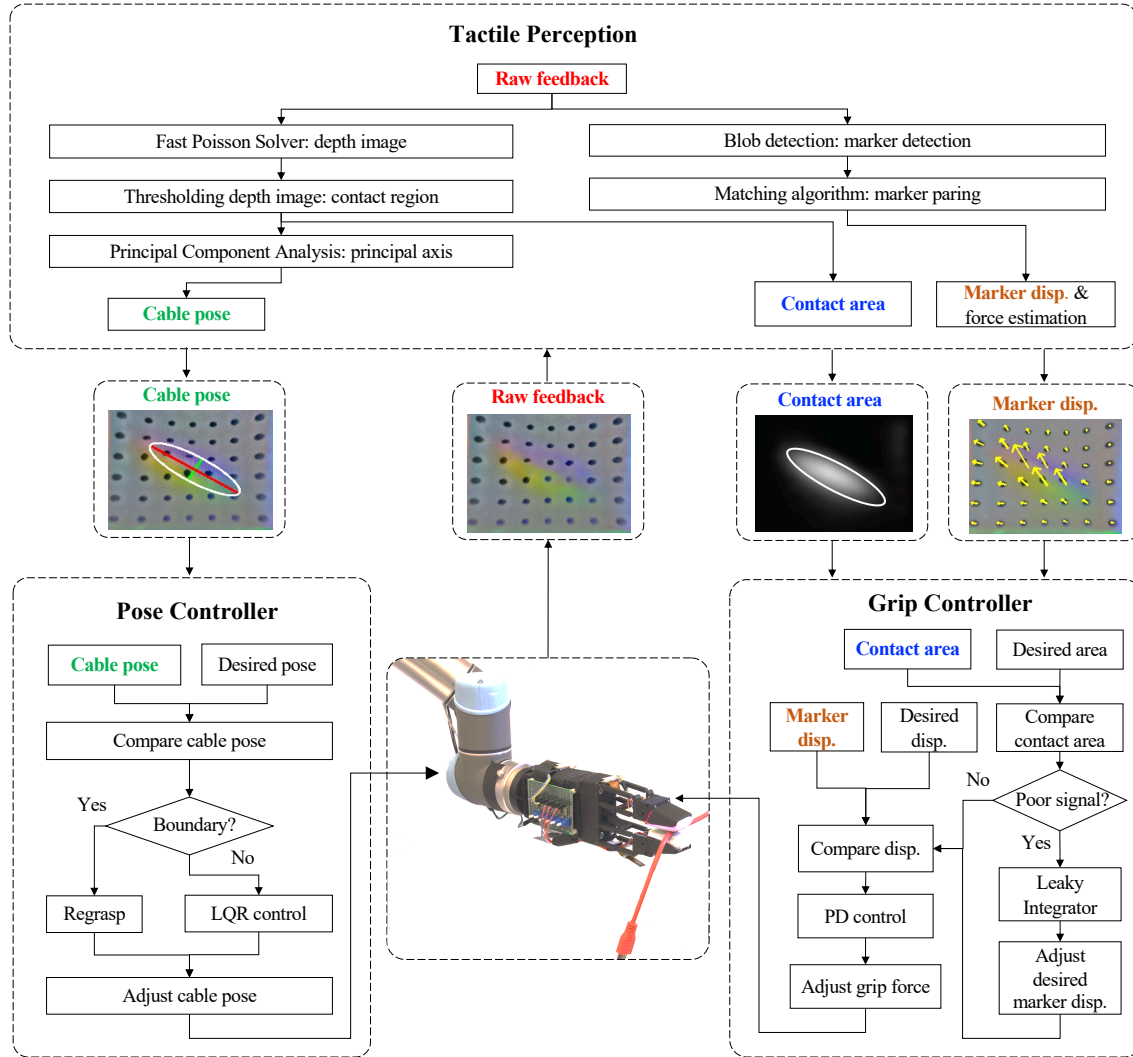


Figure 2-8: **The robotic cable manipulation flowchart.** The flowchart includes three modulus: tactile perception, pose controller, and grip controller.

USB cable with nylon surface; thick HDMI cable with rubber surface; thick nylon rope; thin nylon rope; and thin USB cable with rubber surface.

2.5.2 Cable following experiments

Experimental process The beginning end of the cable is initially grasped firmly with the fixed gripper secured at a known position. The moving gripper picks up the cable and follows it along its length until reaching its tail end. During that process, the grasping force is modulated with the cable grip controller and the pose of the

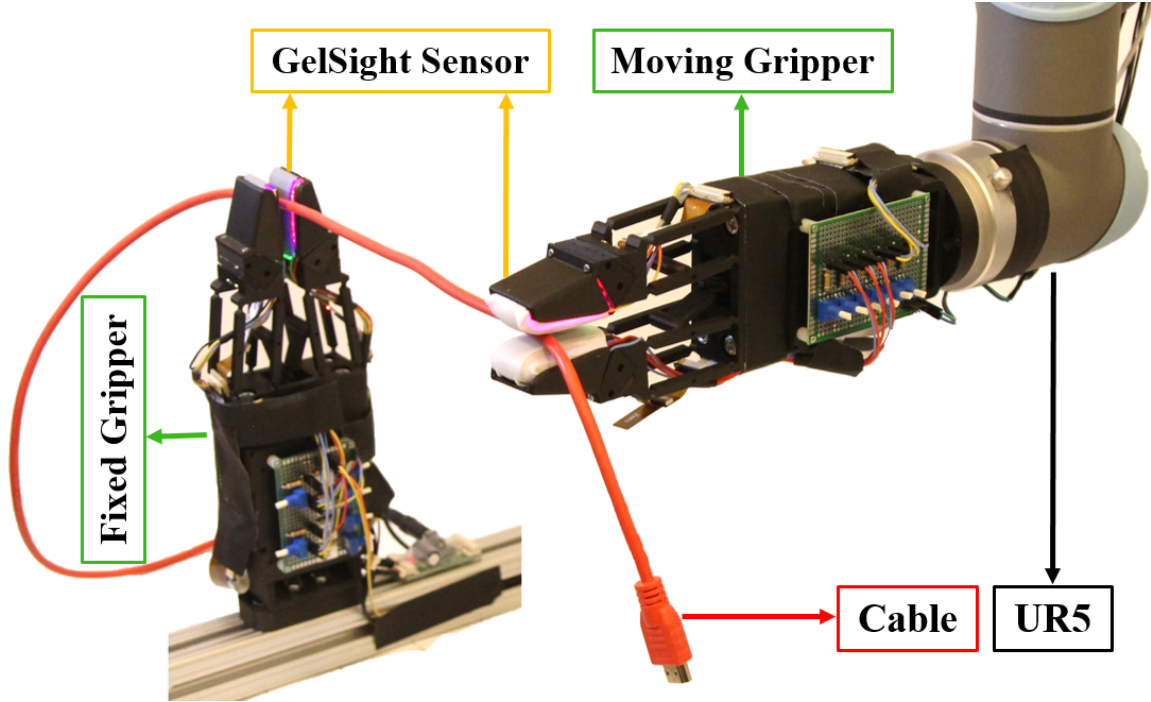


Figure 2-9: **Experimental setup.** UR5 robot arm and two reactive grippers with GelSight sensors.

cable is regulated with the cable pose controller. We convert the control input ϕ to the robot velocity along x and y axis in the world frame with the following kinematic relation:

$$\begin{aligned} v_x &= v(\cos(\phi + \alpha)), \\ v_y &= v(\sin(\phi + \alpha)), \end{aligned} \tag{2.7}$$

where v is a predefined magnitude of the velocity of the robot. The moving gripper can regrasp the cable by feeding the holding part to the fixed gripper if it feels it is going to loose control of the cable, or the robot reaches its workspace bounds. Within a regrasp, the robot adjusts the position of the moving gripper according to the position of the cable detected by the fixed gripper.

Metrics we use three metrics to evaluate performance:

- Ratio of cable followed vs. total cable length
- Distance traveled per regrasp, normalized by the maximum workspace of the moving gripper.

- Velocity of the sliding task, normalized by max velocity in the x direction.

Note that all these metrics have a max and ideal value of 1.

Controller comparison We compare the proposed LQR controller with three baselines: 1) purely moving the robot to the x direction without any feedback (open-loop controller), 2) open-loop controller with emergency regrasps before losing the cable, 3) Proportional (P) controller we use to collect data. Since the initial configuration of the cable affects the result, we try to keep the configuration as similar as possible for the control experiments and average the results for 5 trials of each experiment.

Generalization We conduct control experiments with the LQR robot controller + PD gripper controller to test the performance across 1) different velocities (v): 0.025, 0.045, and 0.065 m/s; and 2) 5 different cables. Similarly, we also conduct 5 trials for each experiment and average the results.

2.5.3 Cable following and insertion experiment

An illustrative application of the cable following skill is to find a connector at the end of a cable to insert it. This is a robust strategy to find the connector end of a cable when it is not directly accessible or under position uncertainties. Here we conduct an experiment on a headphone cable. The relative position in the workspace of the hole where to insert the connector is calibrated. The cable following process is identical to what we illustrated in the previous section. We detect the plug (thick) based on its geometry difference compared to the cable (thin) using the GelSight sensor. To estimate the pose of the plug before insertion, we use the same tactile estimation method as used to estimate the cable pose during cable following.

2.6 Experimental Results

In this section, we evaluate the performance of the linear dynamic model. We then detail the results of the cable following experiments with different robot controllers, different velocities and different cables, according to the evaluation metrics. See

Fig. 2-14 for a summary. We also show the results of the cable following and insertion experiment (Fig. 2-13).

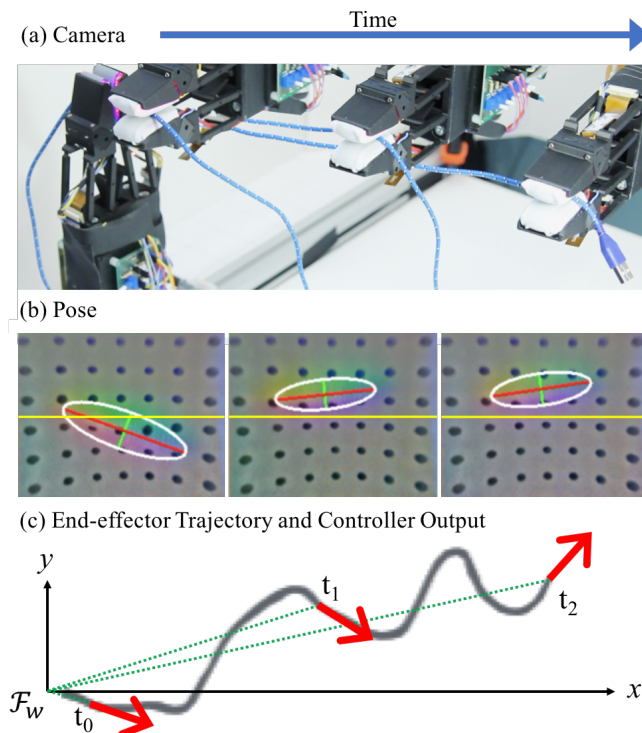


Figure 2-10: **Cable following experiment.** For three instances in time, (a) camera view; (b) pose estimation from tactile imprints, where the yellow line in the center indicates the desired in-hand pose alignment; (c) top view of the trajectory of the end-effector and velocity output of the LQR controller, shown in red. The green dotted line illustrates α . The controller keeps adjusting the cable state in real-time by changing the moving direction to achieve the desired pulling angle.

2.6.1 Linear dynamic model evaluation

We evaluate the learned linear dynamic model with a collected dataset of cable-gripper interactions.

Data collection We collect the data in the similar way as the system runs at the test time, as described in Sec. 2.5.2, experimental process. But we let the fixed gripper always grip the beginning end of the cable, so that the data collection can run repeatedly. The sliding motion can reset the cable to different initial configurations. In addition, we add some perturbances to the initial cable configurations for data di-

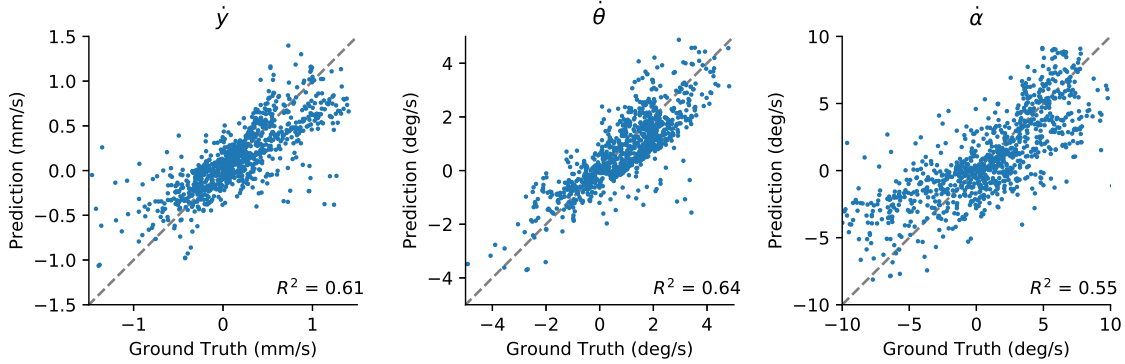


Figure 2-11: **Predicted vs. actual velocity** \dot{y} , $\dot{\theta}$, and $\dot{\alpha}$, of the generalized coordinates of the cable-gripper dynamics, as defined in Fig. 2-7.

versity. We use approximately 3000 data points with a single cable. Each data point contains the measured states y , θ , α , the recorded control input ϕ , and the estimated velocities \dot{y} , $\dot{\theta}$, $\dot{\alpha}$. We train the model with two thirds of the data, and validate the result with the rest.

Evaluation Fig. 2-11 shows the performance of the linear regression model over \dot{y} , $\dot{\theta}$, and $\dot{\alpha}$. The horizontal axis corresponds to ground truth and vertical axis represents the estimated velocities. According to the comparison, the simple linear model captures the cable dynamics with coefficient of determination R^2 of 0.61, 0.64, 0.55 respectively. We further show a sample sequence of prediction results during one sliding in Fig. 2-12. The predicted velocities (orange dash line) align well with the ground truth (blue solid line) most of the time. We expect non-linear models like Gaussian Process and Neural Networks could achieve more accurate predictions, but whether those more accurate predictions would translate to better control performance, would need to be investigated.

2.6.2 Controller evaluation

We further evaluate the LQR controller with the learned linear dynamic model, and demonstrate it is sufficient to accomplish the task efficiently.

We compare four different robot controllers: open-loop, open-loop with emergency regrasps, P controller, and LQR controller. The top row in Fig. 2-14 shows that the

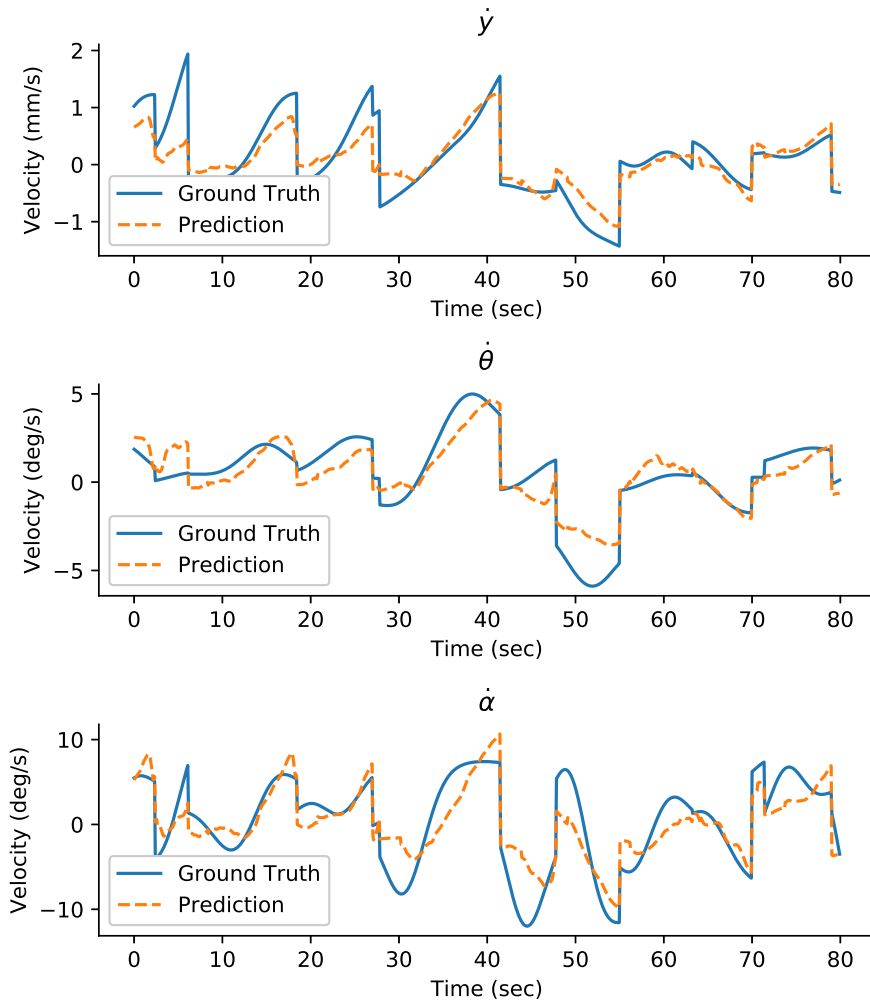


Figure 2-12: **Sequence of predicted (orange dash line) and actual (blue solid line) velocity \dot{y} , $\dot{\theta}$, and $\dot{\alpha}$.**

open-loop controller only follows in average 36% of the total length of the cable. The gripper loses the cable easily when it curves. The simple addition of emergency regrasps is sufficient for the open loop controller to finish the task. This indicates that a timely detection of when the cable is about to fall from the gripper is important for this task. This controller, however, still requires many regrasps and is slower than the P and the LQR controllers. The results show that the LQR controller uses the least number of regrasps compared to other controllers. The LQR controller does not show much experimental improvement in the velocity metric, possibly because the robot travels more trying to correct for cable deviations.

Figure 2-10 shows snapshots of the experimental process using the LQR controller.

Note that this controller always tries to move the gripper back to the center of the trajectory once the cable is within the nominal configuration since α (the angle between the center of the cable in hand to the beginning end) is also part of the state. This can be observed from the middle time instance of Fig. 2-10, where the cable is straight and close to the middle of the GelSight sensor, but α is large (the angle between the x axis and the green line to t_1 in the bottom of Fig. 2-10). The output of the controller shows the direction to the center of the trajectory. The pose in the last time instance is similar, but because α is smaller, the controller outputs a direction that will correct the cable pose. This feature is an advantage of the LQR controller over the P controller.

2.6.3 Generalization to different velocities

The model of the cable-gripper dynamics is fit with data collected with robot velocity of 0.025 m/s. We also test the LQR controller at 0.045 m/s and 0.065 m/s. The results in the second row of Fig. 2-14 show that the performance does not degrade, except requiring more regrasps per unit of distance travelled. This is likely because, going faster, the controller has less time to react to sudden pose changes and, therefore, tends to trigger regrasps more. Although the number of regrasps increases with larger velocity, the total time is still shorter due to the faster velocity.

2.6.4 Generalization to different cables

We also test the system with the LQR controller on 5 different cables, each with different physical properties (diameters, materials, stiffness). In experiments, the system generalizes well and can follow 98.2% of the total length of the cables.

Comparing the performance on the different cables shows that cable 4 (thin and light nylon rope) requires the most regrasps. It is difficult to adjust in-hand pose since it is very light and the un-followed part of the cable tends to move with the gripper. The cable with best performance is cable 5 (thin and stiff rubber USB cable), which is stiff and locally straight most of the time.

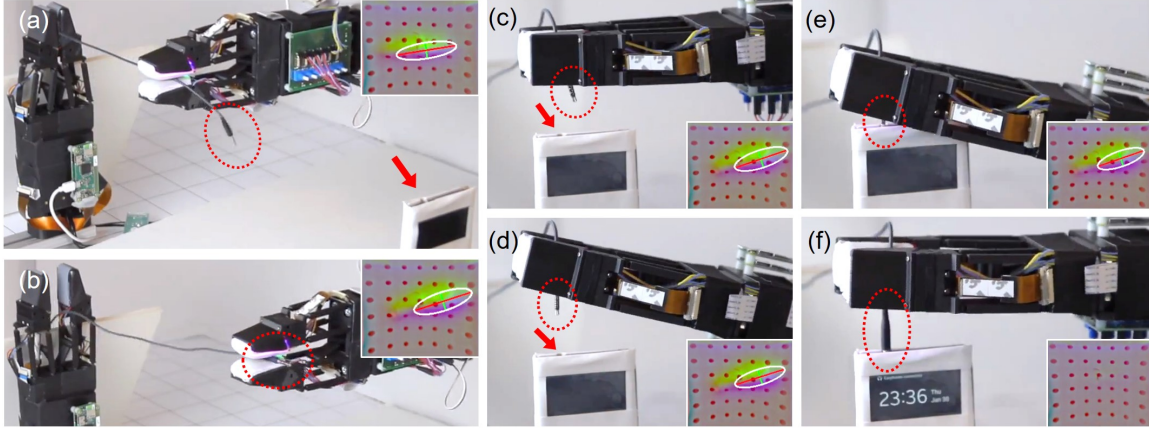


Figure 2-13: **Headphone cable following and insertion process.** (a)(b) cable following to the plug end, (c) plug on top of the hole with pose mismatch (d) plug pose adjusted and aligning with the hole, (e)(f) cable plugged into the headphone jack on the phone. The plug is labeled with red circle and the headphone jack is labeled with red arrow.

The tactile reactive gripper in this paper is designed to manipulate cables in our daily life such as USB cables, earbud cables, ethernet cables, etc. A common feature of these cables is that they are relatively thin. For cables with larger diameter, the contact region will increase, and the depth (cable penetration into gel) will decrease, given the same gripping force. Accordingly, the major axis of the ellipsoid and the minor axis of the ellipsoid will increasingly become of similar magnitude, which will be difficult to estimate the direction of the cable. One potential solution is to scale up the sensor and the gripper for larger diameter cables, or, as humans do, use larger contact areas, such as multiple fingers, or palms.

2.6.5 Cable following and insertion

The process to grasp, follow, and insert the headphone cable is illustrated in Fig. 2-13. Parts (a) and (b) show the robot following the cable all the way to the plug and identifying the moment it reaches the plug. After the plug is detected, the fixed gripper opens and the robot moves the plug over the headphone jack on the phone shown in Fig. 2-13(c).

After cable sliding, the gripper uses the tactile feedback from a GelSight sensor to localize the plug and align it with the hole, as shown in Fig. 2-13(d). Afterwards,

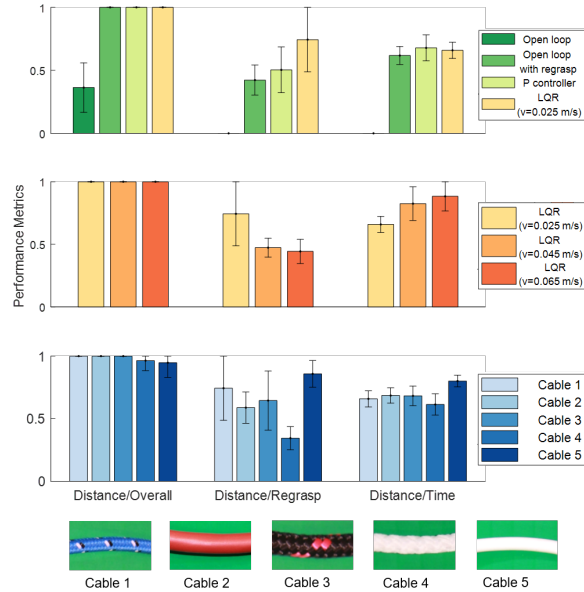


Figure 2-14: **Experimental results.** Different robot controllers (top), different following velocities (middle), different cables (bottom). For visualization, the three metrics are normalized to $[0, 1]$ by dividing 100%, 0.45 m, and 0.02 m/s respectively (max 1, ideal 1).

the robot moves down to insert the cable into the headphone jack in Fig. 2-13(e)-(f). We repeat the insertion only experiment (plug directly fed to the gripper by human with random pose) for 20 times and can insert the headphone plug with an 85% success rate. For better visualization of combined experiments of cable following and insertion, see videos at <http://gelsight.csail.mit.edu/cable/>

2.6.6 Failure cases

We observed several failure cases that deserve further thought.

- **Hand-to-hand regrasps.** Regrasps make it possible to robustly follow cables with arbitrary length. However, it needs the coordination of two grippers, and break the contact temporarily from each of them. One failure mode is that the fixed gripper (currently without touch sensing) fails to grasp the cable after breaking the contact. This is because the piece of cable between two grippers has uncertain orientation depending on its stiffness. Stiffer cables hold straight, and softer ones bend more easily under gravity. Constant relative pose between

two grippers may result in regrasp failure. We could improve the robustness by adding touch sensing to both grippers and applying an iterative regrasp strategy.

- **Plug misdetection.** One failure mode for following-and-insertion is plug misdetection. To detect the end of the cable, the system needs to constantly search for the plug during sliding. Misdetection of the plug can cause early or late stop, which results in a failed insertion. We use the thickness difference between the cable and the plug to distinguish them in the earphone example. We could apply more advanced computer vision methods [88, 67, 8] to increase the detection accuracy.
- **Extreme plug pose.** Another failure mode includes extreme plug orientation (large θ), which could make it impossible to avoid collision during insertion. We could improve it by adjusting the plug pose in gripper before insertion, e.g. with regrasping.

2.7 Conclusions and Discussion

2.7.1 Conclusions

In this paper, we present a perception and control framework to tackle the cable following task. We show that the tight integration of tactile feedback, gripping control, and robot motion, jointly with a sensible decomposition of the control requirements is key to turning the—a priori—complex task of manipulating a highly deformable object with uncontrolled variations in friction and shape, into an achievable task. The main contributions of the work are:

- **Tactile Perception.** Applying vision-based tactile sensor, like GelSight, to cable manipulation tasks. It provides rich but easy-to-interpret tactile imprints for tracking the cable pose and force in real-time. These local pose and force

information about the cables are otherwise difficult to be captured by a vision system during continuous manipulation because they are usually occluded, expensive to interpret, and not sufficiently accurate.

- **Tactile Gripper.** The design of a reactive gripper that uses compliant joints, making it easy to fabricate, and protecting the tactile sensor from unexpected collisions. The gripper modulates grasping force at 60 Hz which enables tasks that need fast response to tactile feedback such as cable manipulation.
- **Tactile Control.** We divide the control of the interaction between cable and gripper into two controllers: 1) Cable Grip Controller, a PD controller and leaky integrator that maintain an adequate friction level between gripper and cable to allow smooth sliding; and 2) Cable Pose Controller, an LQR controller based on a linearized model of the gripper-cable interaction dynamics, that maintains the cable centered and aligned with the fingers.

The successful implementation of the tactile perception and model-based controller in the cable following task, and its generalization to different cables and to different following velocities, demonstrates that it is possible to use simple models and controllers to manipulate deformable objects. The illustrative demonstration of picking and finding the end of a headphone cable for insertion provides an example of how the proposed framework can play a role in practical cable-related manipulation tasks.

2.7.2 Discussions

Robotic manipulation has had an impact on a range of real-world tasks, such as pick-and-place and assembly. In most cases, the objects manipulated are rigid. Manipulation of deformable objects is more challenging since soft materials are represented by more complex states and follow more complex dynamics. A common approach to manipulating deformable objects is to iteratively transition between static stable states via pick-and-release sequences. This reduces complexity, but also makes manipulation inefficient. Natural manipulation of deformable objects observed from humans

involves dynamic interactions such as sliding along an earbud cable to find the plug or sliding along the edge of a sheet to find its corner.

This “sliding” motion yields new challenges: cables are highly deformable with complex dynamics, and the operation requires real-time adjustments. Correspondingly, the design of control policies for this type of task becomes difficult. However, we can exploit the local constraints imposed by the same sliding motion to simplify control, specially when supported by state feedback via advanced tactile sensors.

Global vs. Local dynamics From a global perspective, the state and dynamics of a deformable object are computationally challenging due to their large number of DOFs. Fig. 2-15 (left) shows the global view of an earphone cable and a piece of cloth. However, the state and dynamics are simplified for specific types of interaction. For example when the cable or piece of cloth are in tension, it is easier to control the task of sliding the fingers. The dimension of the deformable object is reduced by the same constraints imposed by the sliding motion. This makes it possible to design simple and efficient controllers for real-time robotic manipulation. In Fig. 2-15 (Right) the state of cable and the cloth is simplified in between the 2 grasping points.

It is worth noting that it would be challenging to apply the current pose estimation method (PCA) to soft cloths because the contact region may no longer be an ellipsoid. One may consider using a different method, maybe a supervised data-driven method, to estimate the principal direction of the edge of the cloth. The same idea applies to very soft cables such as wool. In terms of the pose controller, soft cloths would also be more challenging than relatively stiff cables for robotic manipulation. We expect that leveraging extrinsic dexterity like the gravity of fabric for self-straightening or partial table support with external forces, will be important. The grip controller might also need to be adjusted such that the gripper opens further to better adjust the in-hand pose when required. But in general, the idea of following tactile features (like principal axis/edges) of deformable objects can serve as an alternative motion primitive to alleviate the complexity of state and dynamic modeling.

The cable following technique we demonstrate in this work bypasses the complexities of global state estimation and control by designing policies that rely only on local

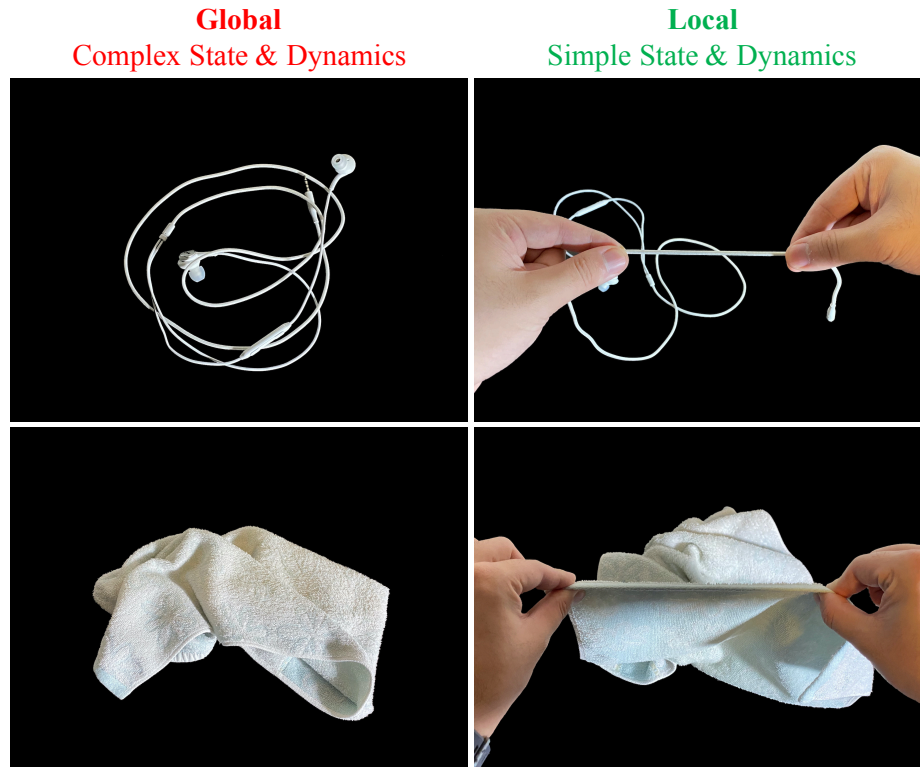


Figure 2-15: **Simplified state and dynamics from sliding.** The comparison of the global (left) and local (right) perspective of manipulating a piece of cable and fabric. From a global view, it is challenging to model the state and dynamics of a deformable objects due to the large number of degrees of freedom. However, the sliding motion adds constraints to the objects, simplifying the local state and dynamics. It enables fast and reactive manipulation skills.

state feedback, which can be captured by tactile sensors. The technique to change the grasp on the cable by sliding the fingers can be thought of as a closed-loop primitive action (the “sliding regrasp”) applicable to a range of objects (not just cables, but also rigid objects and other types of deformable objects, e.g., cloth). On the perception side, this requires a sensor that can track local motions of the local geometry at contact. On the control side, this requires a model of the local pulling-sliding dynamics.

Contact state control & Object state control This work proposes a novel control framework for robotics manipulation of deformable objects with sliding operations by decoupling the complex manipulation policy into two simple independent controllers: contact state control (i.e., cable grip controller) and object state control (i.e., cable

pose control). This is equivalent to the decoupling control approach proposed for tactile dexterity by Hogan et. al [55] with manipulation primitives for rigid objects, and by enforcing sticking. In the case of this work, however, we apply it to a primitive aimed at manipulating deformable objects with sliding interactions. The key idea is the same: one controller (contact state controller) regulates the interaction between the gripper and object to a nominal contact state, and a second controller (object state controller) manipulates the object by exploiting the regulated nominal contact state.

In our case, the contact state controller regulates the contact forces between the gripper and cable to facilitate smooth sliding, while the object state controller maintains the cable at the center of the gripper. These two orthogonal controllers interact and benefit from each other. By decoupling the control policies into two orthogonal controllers, we can use simple controllers (PID and LQR respectively) in separate threads to perform the task, enabling real-time control for complex dynamic manipulation of deformable objects.

Chapter Acknowledgement

This chapter was a joint work with Yu She, Siyuan Dong, Neha Sunil, Alberto Rodriguez, and Edward Adelson. This work was supported by the Amazon Research Awards (ARA), the Toyota Research Institute (TRI), and the Office of Naval Research (ONR) [N00014-18-1-2815]. Neha Sunil is supported by the National Science Foundation Graduate Research Fellowship [NSF-1122374].

Chapter 3

SwingBot: Learning Physical Features from In-hand Tactile Exploration for Dynamic Swing-up Manipulation

In Chapter 2, we introduced the tactile-reactive sliding, which provided an efficient and robust way for cable manipulation. Another way to increase robot dexterity is by dynamic manipulation [100]. The dynamic motion of the robot arm provides extrinsic dexterity [27]. However, the dynamic manipulation tasks are usually sensitive to the physical properties of the objects [169]. Vision alone can be ambiguous, since objects with similar appearances can have very different physical properties [149]. In this chapter, we will apply tactile exploration for learning the physical features of unknown objects under different interactions. The learned features can significantly improve the downstream dynamic swing-up manipulation.

3.1 Introduction

As applications for robotic manipulation shift from industrial to service tasks, the need for robots to deduce the physical properties of objects increases. To cope with

the diversity of objects and tasks in the real world, robots require models that can quickly infer the physical properties of objects, with as few interactions as possible and without explicit supervision. These models could allow the robot to perform more dynamic interactions with its environment or with held objects in the cases where in-hand manipulation is desired. Vision based methods for learning physical object representations through dynamic interaction have shown some promise towards achieving such models [149]. However, vision based approaches are still restricted to interactions in structured environments and do not address the limitations of deploying deep learning based vision systems into the real-world scenarios.

Tactile sensing can be seen as an attractive alternative to vision. In particular, vision-based tactile sensors provide direct observations of the deformation caused by contact with an object [163]. Considering the local nature of these observations, the influence of environmental noise is negligible, making methods developed with this modality potentially more transferable to real-world environments. Additionally, vision-based tactile sensors are able to accurately estimate the normal and shear forces being applied to the sensing surface. So rather than designing an environment to make the influences of external forces easily observable with vision, it is preferable to have very accurate sensing directly at the interaction points, i.e, performing interactions with a sensorized hand. Therefore, tactile sensing seems like an appropriate modality for learning object physical representations. However, it is not without its limitations as these sensors are soft, making the modeling and measuring of the properties of the sensor itself more complex.

In this work we develop a method to infer the physical parameters of an unknown object through in-hand exploration. To do this, we use the information provided by a GelSight sensor [163] to learn a low-dimensional embedding of the object’s properties as well as the properties of the GelSight itself. We learn the embedding in a self-supervised fashion and use it to optimize the performance of a dynamic in-hand manipulation task. In particular we have the robot swing-up a set of unknown objects to a desired pose in-hand. We find the optimal control parameters for the swing-up task with the aid of a swing-up angle predictor that uses our learned embedding as

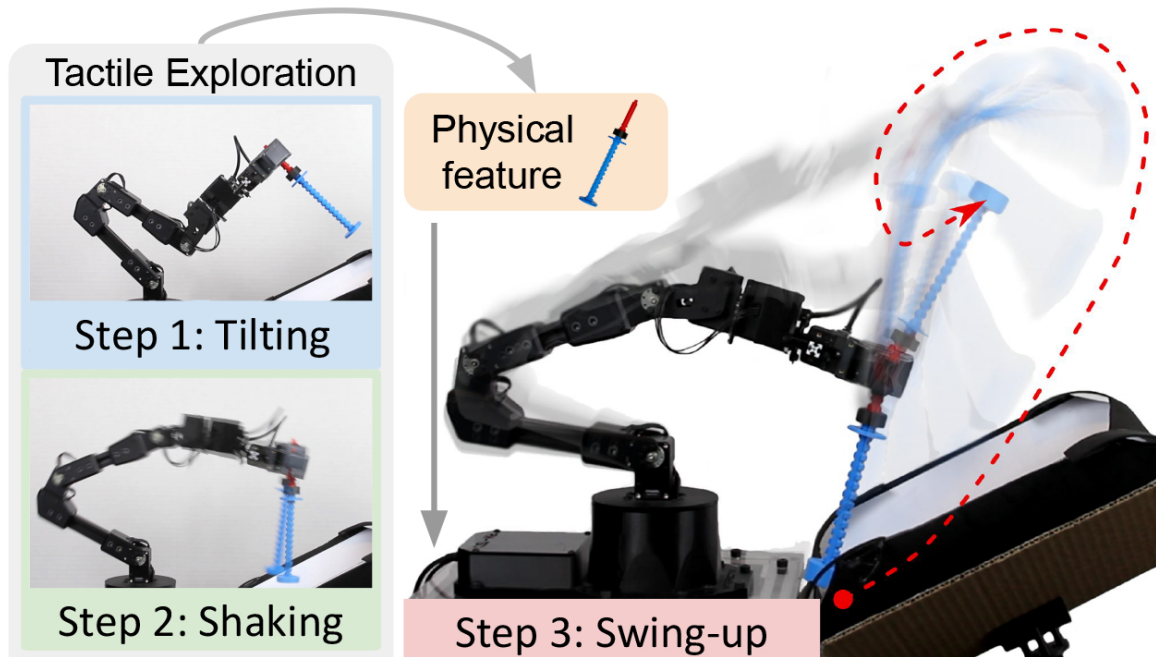


Figure 3-1: **SwingBot**. We develop a learning-based in-hand physical feature exploration method with a GelSight tactile sensor, which assists the robot to perform accurate dynamic swing-up manipulation.

input. We also prove the portability of this embedding to new tasks by showing that we can use it to directly regress to object parameters such as mass, center of mass, moment of inertia and friction.

Our approach consists of two main components: (1) an information fusion model and (2) a forward dynamics model. SwingBot starts by performing two in-hand exploration actions, tilting and shaking. As each of these actions provides different information about the physical parameters of the object, a fusion model takes the information from both actions in order to learn a joint physical feature embedding of the object in-hand. Once the embedding is learned, a forward dynamics model uses the embedding and the control parameters that generate the swing-up motion in order to predict the final swing-up angle.

The main contribution of this work is to demonstrate that the robot is able to learn a low-dimensional embedding of the physical features of a held object from dense tactile feedback acquired through a small number of active exploration actions. The learned embedding allows the robot to accurately and consistently perform a

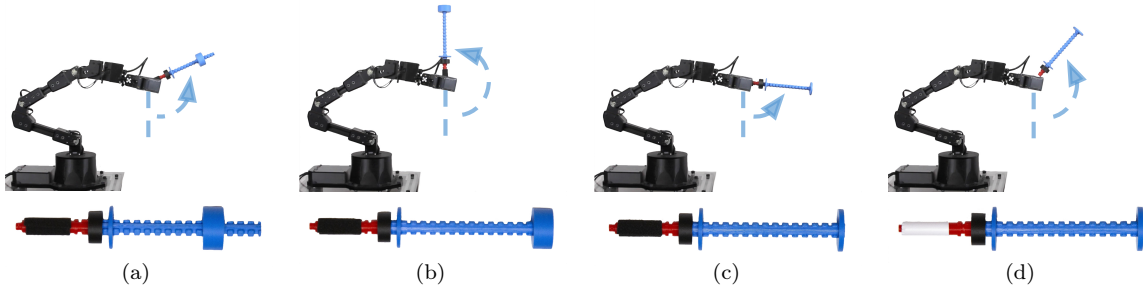


Figure 3-2: **Challenges.** Swing-up is a highly-dynamic process, where changing objects’ physical properties would have a big impact on the final swing-up angle. Here we show, with the same control parameters, that the dynamics vary when the objects vary: same mass but different center of mass (a)(b); different mass (b)(c); and different friction coefficient (c)(d).

swing-up task on a set of objects, with an overall 17.2 degree error on unseen objects. Furthermore, our experiments show that the fusion network can accurately estimate physical parameters of unknown objects once the features are disentangled.

3.2 Related Work

Robotic manipulation has been dominated by the paradigm of kinematic manipulation, and for good reasons. In reducing the effects of task dynamics, it is easier to ensure that a robot can perform its task consistently without error. However, this has limited the application of robotics to a set of simple tasks like pick-and-place. As robotic manipulation becomes more ubiquitous, the need for robots that can perform more tasks becomes important. One path forward is to increase the mechanical complexity of robots by using dexterous hands. However this also comes with a cost in terms of control and design complexity. Alternatively, [100] illustrates that simple mechanical designs can achieve more than pick-and-place if we reconsider the task dynamics.

Inspired by [100], researchers have been successful in developing methods that exploit the task dynamics for performing actions like dynamically sliding an object in-hand [121], tossing an object into the air to regrasp it [27], and swinging up an object to a desired pose [123]. However, these methods require experts to first determine

which parameters of the system are important for the task, a model of the dynamics, and accurate measures of the physical properties of importance for each object used. Therefore, these methods are hard to deploy in real-world environments.

To alleviate the need for careful modelling and accurate measurements, researchers have been working on an alternative method known as intuitive physics [147, 2]. Intuitive physics allows a robot to estimate the parameters of an object via learning based approaches and interaction. In [147, 87, 149, 44], direct regression over the physical parameters of an object, like mass and friction, was performed for tasks like sliding an object and predicting the stability of a tower of stacking blocks. However, knowing exactly which physical parameters are needed for a task or directly observing those parameters from feedback may be difficult. So, several methods [2, 42, 156, 169] instead indirectly estimate object parameters by learning an object embedding in a self-supervised way for tasks like pushing and tossing an object to a desired pose. However, these methods still require a structured environment. In particular, [149] used a set of ramps to make the result of a dynamic interaction easily observable with vision.

Rather than using the environment we can instead use in-hand manipulation to extract properties about the object. In fact, [83] suggest that humans perform a set of exploratory procedures to extract object properties like friction, mass and center of mass. While, it is possible to monitor in-hand interactions with vision, [16] shows tactile sensing outperforms vision alone when doing tasks that require feedback about contact interactions like determining if a grasp is successful. This work along with other works that explore tactile sensing for tasks like slip control [136, 36, 126], re-grasping [21, 30, 56], contour following [85, 53, 119], and ball manipulation [131] focus mainly on static or quasi-static interactions. The object’s physical properties have less of an influence on the performance of a controller for static or quasi-static interactions than they would have in more dynamic manipulation tasks like swing up. As a result, none of aforementioned works that explore tactile sensing estimate the physical parameters of the object. In contrast, we focus on learning physical representations from simple in-hand tactile exploration, and show that such representations

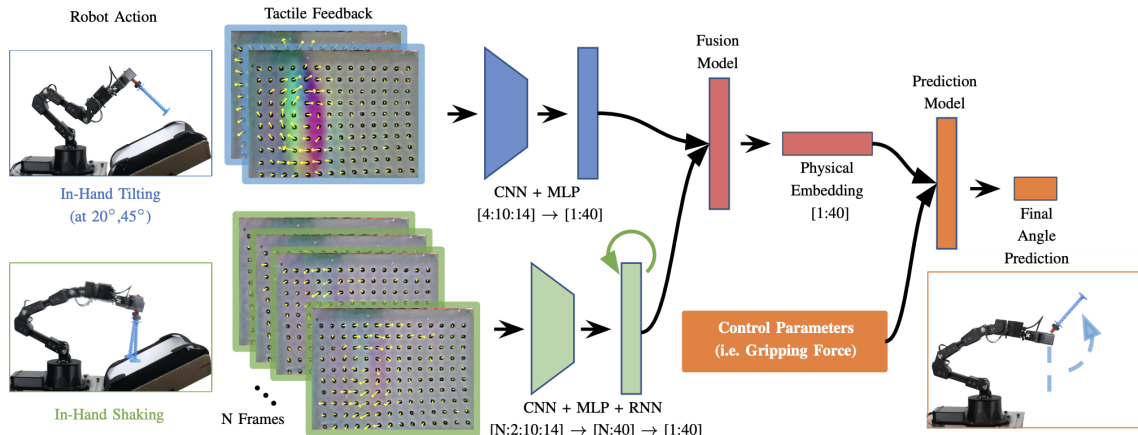


Figure 3-3: **Overview of the architecture.** The robot takes several steps to acquire and use the physical features of the held object: (1) *Tilting* the object at 20° and 45° . The corresponding marker information is encoded by a network with CNN and MLP into a 40-dimensional embedding. (2) *Shaking* the object. The sequence of marker information is processed by a RNN network into a 40-dimensional embedding. (3) A fusion model concatenates the embedding from both actions and outputs a fused physical feature embedding. (4) A prediction model takes the physical embedding and control parameters as input and outputs a prediction of the final swing-up angle. **During training**, the whole pipeline is trained in an end-to-end fashion using the final angle for self-supervision. **During inference**, a set of control parameters are uniformly sampled. The action with the prediction result closest to the goal is selected to perform the swing-up.

are useful for manipulation tasks that requires physical knowledge.

3.3 Method

The goal of SwingBot is to enable the robot to swing up an unknown object to a desired pose ($0^\circ \sim 200^\circ$) after performing a single exploratory action. In [123], the authors suggest the robot must first build a dynamic model of the task, and once the robot has a notion of what this model is, it must then extract which physical parameters of an object are keys to completing the task. Thus, when a novel object is introduced, the system only needs to extract those parameters to tune the model. Therefore, we create a method to estimate the desired control parameters of a hand coded control policy by performing a set of hand coded exploration actions. To accomplish this we use GelSight, a vision based tactile sensor, to monitor the state of

the object while performing in-hand exploration of the object in the form of shaking and tilting. These exploratory procedures extract different type of object information, and as a result we create a method to fuse the information from both procedures into a joint physical feature embedding of the object. We then create a forward dynamics model that uses the embedding to infer which action will result in our desired object pose.

3.3.1 GelSight

While previous methods exploring physical object property estimation monitored the result of a dynamic interaction with vision [149], vision as a modality has its limitations for this task. Beyond errors in state estimation due to environmental noise, it lacks the ability to perceive the forces being applied to an object. Hence, if you were performing an exploratory action like tilting an object in-hand to estimate its mass, its change in position as you tilt the object would be almost imperceptible, as seen in Fig 3-4. Therefore, we rely on tactile sensing, the GelSight sensor [163] in particular. The GelSight enables us to have high resolution information about the contact surface between the object and the finger. This enables us to have information about local geometry of the object for pose estimation. Beyond that the sensor used in this experiment is equipped with markers along the sensing surface which provides information about tangential displacements, giving us rich information about the shear forces and torques being applied to the sensing surface.

3.3.2 Information Fusion for Multiple Exploration Actions

While the use of a GelSight has its advantages in providing rich information about the contact dynamics between the finger and the object, it also comes with its limitations. The material used to make the GelSight (Polydimethylsiloxane) exhibit nonlinear mechanical properties that are difficult to measure and model. So, while previous approaches [149] were able to directly regress over physical properties like mass and friction and perform a forward simulation, we take a different approach. Rather

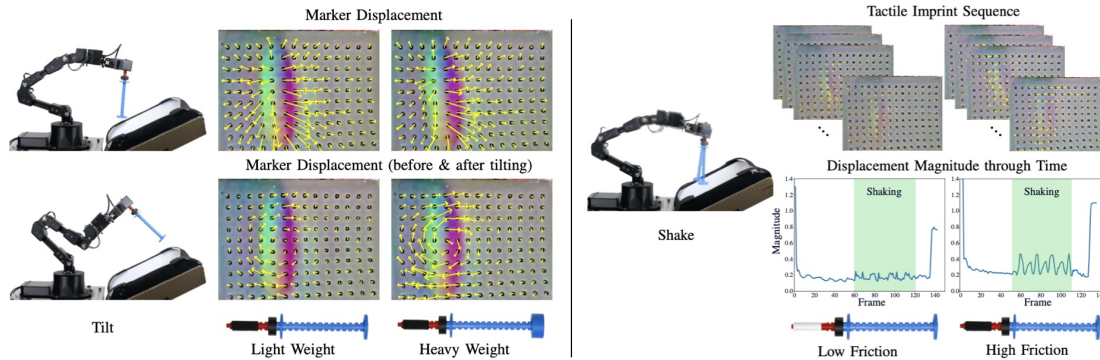


Figure 3-4: **Exploration actions and GelSight Signals.** The robot executes two in-hand explorations, tilting and shaking, to acquire tactile observations of the object. When tilting, different force and torque distributions are generated by the objects weight can be observed. When shaking, different frictions and vibrations can be observed from temporal sequences of tactile signals.

than regressing over the physical parameters, we hand design a set of exploratory procedures that clearly encode physical properties of the object like fiction and mass, and then let the model create its own low-dimensional embedding of the object using self-supervised learning in hopes it also encodes the gel’s dynamics.

In designing these exploration actions, we had to determine what set of parameters to search for. In [123], a dynamic analysis of the swing-up task was performed, concluding that the **surface friction**, **mass of the object**, **center of mass** and **moment of inertia** play roles in swing-up dynamics. Since we use a parallel gripper for this task we are limited to what we can choose in terms of actions. We determined that shaking and tilting the object in hand were the only methods that can be performed reliably. After some experimentation with these behaviors, we determined that tilting was able to give us information about **mass**, **center of mass** and the **moment of inertia**, while shaking is able to inform us of the **friction** of the object as show in Fig. 3-4.

In-hand Object Tilting: Using tactile feedback and tilting the object in-hand to different angles provides us information about the **mass** and **center of mass**, as showed in Fig. 3-4. We observed that as the object was tilted to a low angle we could obtain its mass, while tilting the object to a larger angle gave us information about the torque being applied to the sensor. Combining mass in torque estimates we were

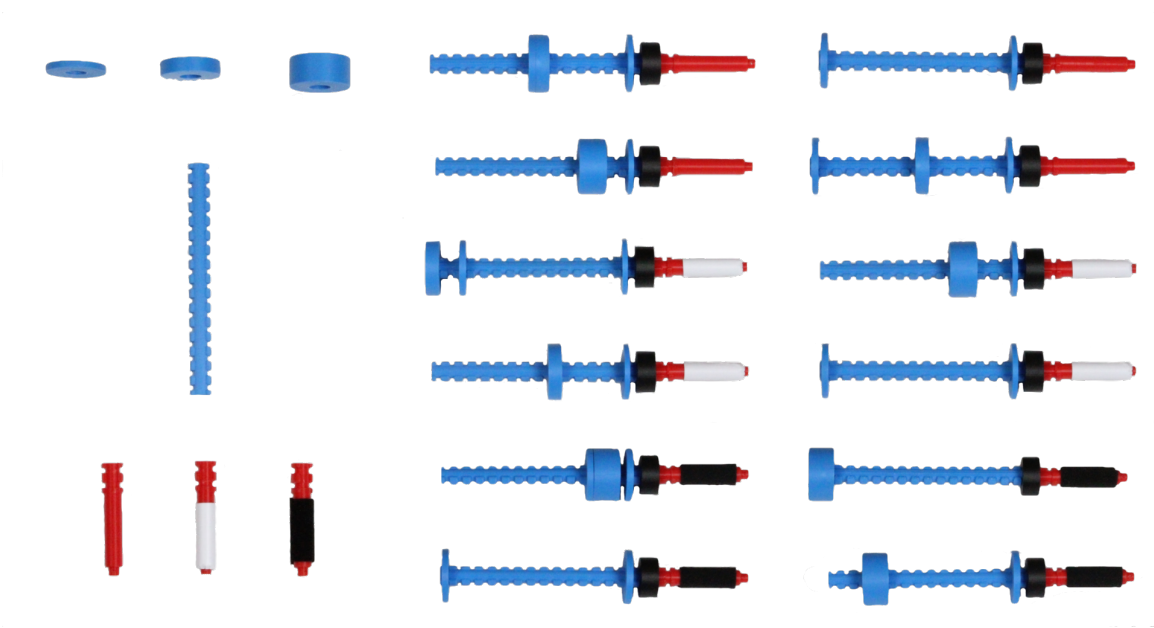


Figure 3-5: **Template objects.** The template objects consist of three components: handle, rack and weights. Different components can be assembled and replaced easily, which creates a variety of objects with different physical properties.

then able to infer the center of mass. Therefore, after the robot grasped the object and held it in-hand, it is then able to tilt the object into 20 and 45 degree poses. The marker feedback ($W \times H \times 2$; $W = 14, H = 12$ in our experiments) from the GelSight tactile sensor at each angle is recorded and used as the input information to the model. Then, the model concatenates the marker information of three angles into a 4 dimensional inputs, followed by a CNN network with kernel size of 5×5 , 3×3 and 2×2 . The last layer of the network is a fully-connected layer which outputs a 40-dimensional feature as the fusion of the learned physical proprieties for tilting.

In-hand Object Shaking: Shaking in turn contains information about the **friction**, and potentially for the **moment of inertia**. After holding the object in a 0-degree pose, the robot first loosens the gripper force to enable a small range of rotation flexibility (Fig. 3-4). Then, the robot starts a quick switch between forward and backward rotations (5 degrees in our experiments) on the joint of the end effector. During this process, we record a sequence of the tactile marker displacements (60-70 frames per trial). Each frame is then processed into a 40-dimensional embedding with the CNN network, which has the same architecture as the one introduced above. Since

we want to extract the inter-frame information of the shaking action, we use a long-short term memory (LSTM) [54], which starts with zero hidden states and iteratively processes all the embeddings of each frame. The last layer will concatenate the hidden states h and cell state c into a 80-dimensional embedding as the fusion of the learned physical properties for shaking.

3.3.3 Prediction Model for Forward Dynamics

To perform the swing-up action we use an impulse-momentum method [3]. The first stage of the swing-up action begins by having the robot build up the object’s linear and rotation momentum by simultaneously accelerating the object upwards and rotating the wrist in the direction of the swing while holding the object firmly. After a short period the robot creates an impulse, by quickly accelerating the object downwards and rotating the wrist in the opposite direction of the swing. At the moment of the impulse the robot loosens the gripper, so that the inertia of the object can overcome the forces of rotational friction and gravity. Thus, the object freely rotates in-hand. After some time, the gripper is tightened to stop the motion of the object at some pose. We use current based position control for the gripper so that the robot automatically decides the gripper width for holding different objects tightly with the same motor torque. When designing the action, the linear and rotational movements of the arm are predefined as well as the timing of gripper tightening, but the robot selects how much the gripper loosens at the impulse. This allows the robot to control the objects deceleration so that it can precisely control the object’s end pose.

In order to use the learned physical features to find the control parameter, the gripper width, for the swing-up manipulation, we propose a forward dynamics model that takes the fused physical features and the action as inputs and outputs a prediction of the final swing-up angle (Fig. 3-3). The model is trained in a self-supervised learning fashion. The data collection is introduced in Sec. 3.4. During the inference mode, the robot first records the marker information of the tilting and shaking actions. Then, the trained information fusion model processes these inputs into a joint physical

feature. After that, a set of gripper widths are uniformly sampled. The prediction model predicts the final swing-up angle for each sampled gripper width and then selects the one with the prediction result closest to the goal pose.

3.3.4 Template Objects and Dataset

When it comes to model generalization, the diversity of the training conditions highly influences the models performance on unseen objects. To this end, inspired by [7], we design a modular system to quickly build a set of test objects. Our object templates are shown in Fig 3-5. There are three major components: handle, rack, and weight. They can be assembled or replaced by simple rotational press-fit. The goal is to change the object’s physical properties easily by placing different weights in different positions and exchanging handles.

With our template objects, we collect a dataset that contains 33 different objects and each object was used in 50 swing-up trials, performed with a random control parameter. These objects contain variance in different category of physical proprieties:

- 3 different **surface frictions** on the handle: foam, slick tape, and plastic.
- 3 disks with different **mass**: 3.7 g, 7.3 g and 14.5 g.
- a pole-shaped rack (15.6 g) allowing for different placement of the disks for variance in **center of mass** (77-134 mm) and **moment of inertia** ($0.03\text{-}0.58\text{ g} \cdot \text{m}^2$)

In each data collection trial, the robot first grasps the object and holds it a 0-degree pose. It then rotates its end effector into two angles (20° , 45° in our experiments), as introduced in Sec. 3.3, and records the marker information from the tactile sensor. After that, the robot resets the object pose to 0° and loosens the gripper force before it starts shaking as introduced in Sec. 3.3. The marker sequence is recorded. Then the robot selects a random control parameter and starts its swing up. The final angle in the end of the swing-up is saved as the supervision ground truth for training the prediction model. At the end of each data collection trial, the robot opens the gripper

and lets the object fall into a recycle box at the bottom of the system. The recycle box will return the object to the same initial position every time so that the robot can automatically start another trial. The reset process is demonstrated in our video supplementary files.

3.4 Experiments

In the experimental section, we would like to answer the following questions: (1) How does the prediction model with the learned physical features compare to the one without physical exploration? (2) How does the fusion of the multiple exploration actions compare to each individual action? (3) Can the physical properties of an object be regressed from the learned features? (4) Are objects with similar dynamics close in the embedding space? (5) Can our method accurately swing-up a set of unknown objects to a desired poses consistently?

To answer these questions, we evaluate our method on both **seen** and **unseen** objects with a 5-DoF robot arm. Here, “**unseen**” refers to objects with physical properties that never appeared in the training set. To assess what information is included in the learned physical feature embedding, we conduct an experiment to directly regress the physical properties (friction, mass, center of mass and moment of inertia) from the physical embedding on both **seen** and **unseen** objects.

3.4.1 Experimental Setup

Dataset for seen objects: We collected data with 33 objects with different physical properties as introduced in Sec. 3.3. For the experiments on “seen” objects, we split the data of each object into 90% for training and 10% for evaluation. Thus, the training set contains 1485 samples (33 objects) and the testing set consists of 165 trials (33 objects).

Dataset for unseen objects: For the “unseen” objects, we split the 33 objects into 27 objects for training and 6 objects (showed in Table. 3.3) for testing. The testing set is composed of a combination of 2 different frictions and two different masses placed

	Rand.	None	PP	Tilt.	Shak.	Comb.
Seen	66.7	25.4	11.0	13.3	10.9	10.2
Unseen	66.7	26.8	18.5	17.6	15.0	12.9

Table 3.1: Quantitative evaluation results of the prediction model with physical embedding from different variants of the fusion model on seen and unseen datasets. The results are shown in degrees.

at 2 different locations. The training set contains 1350 samples and the testing set consists of 300 trials.

Architectures: We compare five model variants that show case the effectiveness of our design choices:

- *None*: No tactile exploration information is given. The model takes the action as input and directly predicts the final swing-up angle.
- *PP*: The numerical value of each physical property (friction, mass, center of mass and moment of inertia) of the object is given to the model as inputs.
- *Tilting*: The model only process the tactile information of the tilting action into the physical features.
- *Shaking*: The model only process the tactile information of the shaking action into the physical features.
- *Combined*: Both tactile information of tilting and shaking actions are processed by the fusion model into a joint physical feature embedding.

Robot experiment setup: As shown in Fig. 3-6, we use a 5-DoF robot arm (ReactorX 150 Robot Arm, Interbotix) for our experiments. For better performance, we replace all the servo motors with DYNAMIXEL XM-430-W350T, ROBOTIS. We use OpenCM9.04 C micro-controller for controlling the robot. In order to get consistent performance, we found that it was critical to send the trajectory to micro-controller buffer in advance and execute on board. Otherwise, the communication latency between PC and micro-controller can produce prohibitively large amounts of actuation noise.

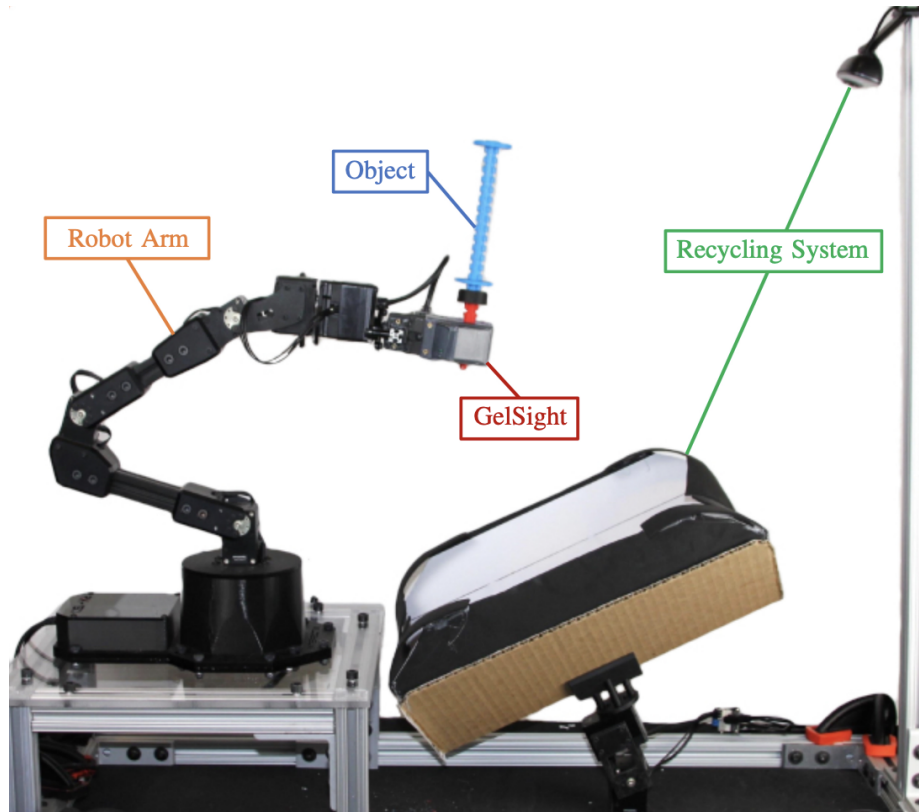


Figure 3-6: **Experiment setup.** The GelSight tactile sensor is mounted on a gripper of the robot arm. The recycling system enables automatic data collection.

3.4.2 Model Performance

Table. 3.1 shows the evaluation results of five of our model variants on both seen and unseen datasets. The metric is the error in degrees on the final angle prediction results. Since the baseline method *None* does not have any information about the physical properties of the object, it could only output a mean value of the training dataset, showcasing the worst performance. On the other hand, the *Combined* method which uses the fusion model to combine the information from both exploration actions achieves the best results, which surpass the *None* for more than 13° on both datasets. This improvement shows the importance of in-hand physical exploration for dynamic manipulation tasks like swing-up.

Also, the *Tilting*, *Shaking* and *Combined* methods outperforms the *PP* baseline method by up to 5 degrees. This is due to the components of the ground truth information being based on the ideal physical model, which has the risk of missing

	Seen			
	Friction	Mass	Cent. of Mass	Mom. of Iner.
Random	33.3%	0.333	0.333	0.333
Tilting	89.6%	0.101	0.150	0.090
Shaking	96.9%	0.121	0.203	0.184
Combined	94.8%	0.085	0.135	0.112
End-to-End	98.9%	0.078	0.083	0.056
	Unseen			
	Friction	Mass	Cent. of Mass	Mom. of Iner.
Random	33.3%	0.333	0.333	0.333
Tilting	75.6%	0.184	0.086	0.141
Shaking	90.1%	0.263	0.125	0.233
Combined	93.9%	0.200	0.099	0.117
End-to-End	95.4%	0.073	0.110	0.095

Table 3.2: Quantitative evaluation results of the physical feature disentanglement on both seen and unseen datasets. The metric for the friction is classification success rate (3 classes). The metric for the rest properties is error in percentage (normalized to 0-1 with the minimum and maximum of the value).

other physical features that also contribute to the model performance such as the elasticity of the contact area of the gripper and the pose of the object in-hand. Since the GelSight tactile sensor provides rich contact information on the finger tip, methods relying on this information have the potential to learn their own joint physical understanding about the held object and the system. These results showcase the advantage of using an intuitive physics reasoning compared to manually engineering physical features.

In the ablation study between *Tilting*, *Shaking* and *Combined*, the performance of the methods with individual exploration action is inferior to the combined version, especially for the unseen situation. Hence, we conducted an additional experiment to evaluate what information is learned in each exploration action and why the combined version could achieve the best performance.

3.4.3 Physical Feature Disentanglement

We use a three-layer MLP as a disentangle network which takes the physical feature embedding as input and regresses to the numerical values of mass, center of mass

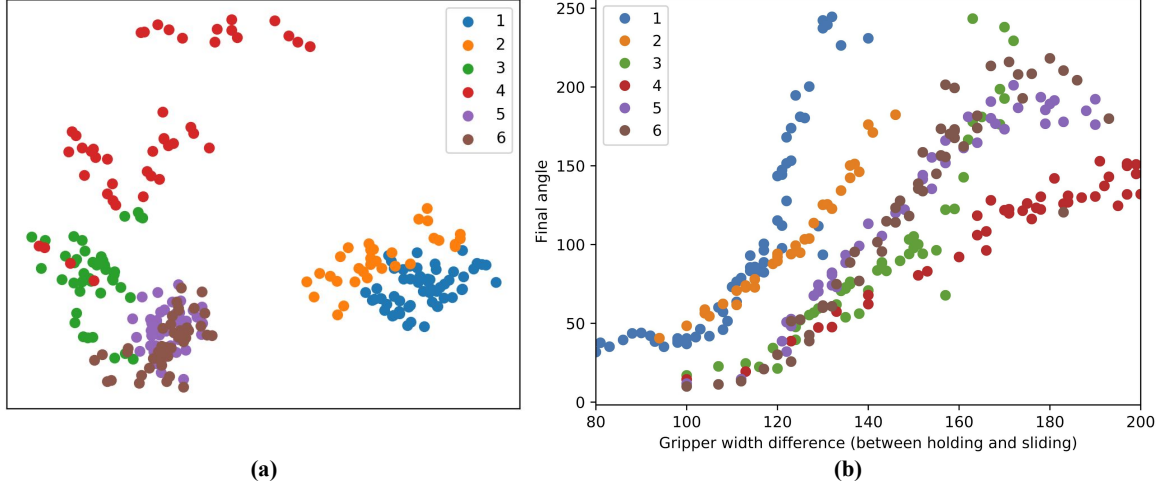


Figure 3-7: **Task-oriented physical feature visualization:** (a) Visualization (with PCA) of the outputted physical embedding (*Combined*) on the testing samples of the 6 unseen objects (listed in Table. 3.3). (b) Visualization of the data distribution (X-axis: control parameter; Y-axis: final angle) of the testing samples of each object. Each color point refers to one data sample. Objects with similar dynamics are also close to each other on the learned physical embedding space (e.g. 5 and 6). And objects with different dynamics are far away from each other (e.g. 1 and 4).

and moment of inertia. Another branch of the network outputs a classification result for the friction (3 classes). The training and testing data for both seen and unseen situations follow the same setting as the prediction model. The weights of the network that generates the physical embedding are fixed and only the disentangle network is trained and tested. In addition to the model variants introduced previously, we add another *End-to-End* method which trains the whole pipeline to output the physical properties. This method can be regarded as the best performance that the model can reach.

Table. 3.2 shows the experimental results of the physical feature disentanglement. The metric of the friction is the classification success rate. The metric for the rest of the physical properties is the error in percentage, where each property is normalized to 0-1 based on the minimum and maximum value. For both seen and unseen situation, all the model variants outperform the *Random* baseline, which proves that all of the physical properties are included in the learned embedding.

In addition, we can observe from the results the difference in focus between each of

the exploration actions. For instance, the *tilting* action is good at reasoning the mass and center of mass, which surpasses the *shaking* for 8% in mass and 4% in center of mass on the unseen situation. This is mainly because the tilting action provides stable torque force signal by placing the held object at different angles, for which is easier to calculate these properties compared to *shaking*. On the other hand, the *shaking* action achieves 93.9% friction classification success rate which is higher than *tilting* action by 15%. This is to the fact that, as opposed to *tilting* when the object is held firmly, *shaking* loosens the gripper to enable in-hand sliding of the object, capturing friction information. Because of the loosening of the gripper, *shaking* failed to acquire the mass and center of mass information, which requires stable observations. It is surprising to find that the moment of inertia of the *tilting* also outperforms the *shaking*. One of the possible reasons for this is the model inferring the moment of inertia based on its understanding of mass and center of mass. The *combined* method successfully fuses the information from both actions and achieved a balanced performance among all the physical properties. This experiments show the importance of fusing multiple exploration actions and why the *combined* method could reach the best prediction results.

3.4.4 Task-oriented Physical Feature

Another advantage of learning joint physical features compared to estimating each property individually is its potential to generate task-oriented feature embeddings, where the objects close to each other in the embedding space can share similar control policies. We use PCA [145] to project the learned physical embeddings from the *Combined* method to points on a 2D plot and visualize all the testing results of the 6 unseen testing objects in Fig. 3-7(a). We also visualize the data distribution of these test samples in Fig. 3-7(b), where the X-axis refers to the control parameter and the Y-axis is the final swing-up angle. As we can see, for objects with similar policy distribution (objects 5 and 6), the distance between their embedding is also short. And for objects with large differences in the policy distribution (objects 1 and 4), the distance between their embedding is large. This result confirms that







ID	Objects	Errors ($^{\circ}$)	ID	Objects	Errors ($^{\circ}$)
1		21.4	2		12.3
3		19.8	4		8.3
5		18.3	6		23.4
Mean	17.2				

Table 3.3: Swing-up results on 6 unseen testing objects (with ID 1-6 same as Fig. 3-7). The robot uniformly samples a set of actions and selects the one with the prediction result closest to the final goal to perform the task. In this table, each object is tested 20 trials (5 trials for each desired angle: 45° , 90° , 135° and 180°) and the mean error is listed.

the learned physical embeddings are indeed task-oriented, which largely benefits the dynamic swing-up manipulation.

3.4.5 Swing-up Results

We deploy the trained model of the *Combine* method in the robot arm. Given the target angle, the robot samples different control parameters, and chooses the one whose prediction is closest to the target angle. We test 20 times (5 times each for 45, 90, 135, 180 degrees as target angle) for each unseen object. The robot is able to adapt the control policy automatically for objects with different physical properties. The evaluation shows that the model performs better on lighter objects which have less uncertainty compared to heavier objects. The detailed results are shown in Table. 3.3.

3.5 Discussion and Future Work

We have presented SwingBot, a robot system that identifies physical features of held objects from tactile exploration, providing crucial information for a dynamic swing-up manipulation. SwingBot is based on a novel multi-action fusion network that combines the information acquired via multiple exploration actions into a joint embedding space. The whole pipeline is trained in an end-to-end self-supervised manner. We used the performance of the swing-up task to compare our embedding with variants

trained with single actions and with swing-up actions that do not consider any form of tactile information. These comparisons showed that swing-up actions that relied on our fusion method achieved the best performances. Furthermore, we showed that the learned task-oriented feature embedding could also be used to successfully regress individual physical properties such as mass, center of mass, moment of inertia and friction.

Current limitations are inherently coupled to the fact that our analysis of the embedding is based on the performance of a single task. This task is very specific and heavily conditioned by the available hardware. The robot platform that was used suffers from high actuation noise increasing the error of the swing-up angle predictions.

In addition, while the GelSight sensors provide very rich information, the current sensing latencies prevent the observation of the full swing-up movement. Using a more robust robotic system in conjunction with a GelSight sensor with lower latencies would potentially enable the use of real time feedback control as opposed to the open loop solution that was proposed.

Regarding future work, one interesting direction is to learn the optimal exploration actions by using the quality of the resulting embeddings to guide the learning. Another interesting direction is to assess how useful these embeddings are for other task and if an embedding learn for one task can be transferred onto other tasks.

Chapter Acknowledgement

This chapter was a joint work with Chen Wang, Branden Romero, Filipe Veiga, and Edward Adelson. Toyota Research Institute (TRI), and the Office of Naval Research (ONR) [N00014-18-1-2815] provided funds to support this work.

Chapter 4

Tactile-Enabled Roller Grasper

In Chapter 2, and Chapter 3, we improved dexterity by combining tactile signals with a robot arm motion, such as sliding and dynamic swing-up. This chapter will explore another direction towards dexterous manipulation by improving in-hand tactile dexterity. Since complex humanoid robotic hands can be challenging to control, we explore simplifying the hand design using a joint motion beyond humans: rolling. We design a roller grasper with tactile sensing, and develop a perception system to tackle the challenges brought by continuous rolling. We will demonstrate its capabilities to manipulate various objects robustly, and use manipulation for more efficient perception. The combination of active roller and tactile sensing opens up a new range of manipulation skills.

4.1 Introduction

Similar to how the dexterity of human hands allows us to accomplish a variety of everyday tasks, the in-hand manipulation capabilities of robots are necessary to accomplish a wide range of complex tasks in different environments. Out of all the manipulation tasks a human hand could perform, in-hand manipulation requires the most dexterity. However, while having high dexterity is desirable for robot hands, designing robot hands or fingers based on their human counterparts might not be optimal. On one hand, while linkage-based finger designs reminiscent of human fingers

are very popular, it is very difficult to obtain similar performance due to limitations such as force density and sensing fidelity. On the other hand, even if human hands are perfectly replicated on robots, they are not necessarily optimally suited for a variety of in-hand manipulation tasks.

Specifically, for in-hand manipulation involving large translation or reorientation, linkage-based robotic hands need to have their fingers repeatedly establish contact, break contact, and re-establish contact for the grasped object to be moved. Such methods could be highly inefficient and difficult to control, motivating a robot grasper with the ability to proficiently manipulate objects within the hand.

The Roller Graspers [160, 161] introduced an entirely new way to manipulate objects within hand. They have demonstrated success in various in-hand manipulation applications. However, one of the major limitations is the lack of local contact information. Tactile sensors are a crucial component for successful robot in-hand manipulation, much like how humans rely extensively on haptic feedback during manipulation tasks. Previous works have shown the importance of tactile sensing in robot in-hand manipulation for linkage-based robot hands [158, 41, 119]. Compared to regular linkage-based fingers, tactile information can be even more important for Roller Graspers for two reasons. (1) Steerable rolling mechanisms introduce more complex contact mechanisms between the rollers and the grasped objects. Understanding the nuances of the contact conditions would further improve the in-hand manipulation capabilities of the Roller Graspers. (2) The steerable rollers may be slightly larger relative to some linkage-based fingers, which makes object occlusion a more prominent issue for external vision-based tracking techniques. As important as tactile sensors can be, none of the previous robot hands with active surfaces successfully incorporated tactile sensors. This is because the continuous rotation mechanism makes it difficult to deploy wires to the surface of fingertips, making it infeasible to integrate traditional tactile sensors based on resistance, capacitance, and piezoelectricity [28, 158, 26]. To overcome this challenge, we chose to integrate a vision-based tactile sensor where the sensing areas are not connected to the sensor module (i.e. camera), allowing tactile sensing on a continuously rotating sensing area.

In this work, we propose a highly non-anthropomorphic robot hand - the Tactile-Enabled Roller Grasper (TERG) - which is a two-finger grasper with steerable active rollers at the fingertips. Each roller is equipped with a vision-based tactile sensor allowing the grasper to obtain high-fidelity contact information during in-hand manipulation. We also demonstrated how the raw tactile information can be processed and used for various tasks. The unique interactions of the steerable fingertips with the contacted objects in combination with the high-definition vision-based tactile sensor allows both successful in-hand manipulation of a diverse set of objects and geometric property identification, leading to tactile SLAM [43, 128].

4.2 Related Work

4.2.1 Robot Hand for In-Hand Manipulation

For the past century, there have been a number of robotic hands designed with in-hand object manipulation capability [111]. Most of these hands are linkage-based, including anthropomorphic hands [25, 32, 52] and other fully-actuated hands [115, 69]. While they may theoretically have sufficient degrees of freedom (DoF) to achieve the dexterity required for in-hand manipulation, the resulting grasp gaiting motions can be difficult to achieve and are an inefficient way to perform in-hand manipulation. Alternatively, in-hand manipulation can be achieved without intentionally switching contact locations, but this severely limits the object’s range of motion. There have also been works that use underactuated linkage-based hands [96, 11] to achieve in-hand manipulation. While these works managed to achieve in-hand manipulation with the limited controllability of underactuated hands, the same issues as with the linkage-based hands still exist as described above.

Another approach towards in-hand manipulation is to use non-anthropomorphic hands [98, 102]. One particular type of non-anthropomorphic hands use active surfaces that allow the hand to move a grasped object with minimal modification of the grasp pose. Earlier works on this type of hands have fixed conveyor direc-

tions [31, 49, 133, 97, 74]. [160] and [161] further developed this concept and incorporated steerable rollers for more dexterous in-hand manipulation. However, as mentioned before, the challenge of incorporating traditional tactile sensors into active surfaces made us choose a vision-based tactile sensor for this application.

4.2.2 Vision-based Tactile Sensing

Vision-based tactile sensors [143, 4, 117, 150, 79, 109, 112] are a type of tactile sensor which converts contact signals into images. It has become increasingly popular in recent years because it provides high-resolution, force-sensitive data that are low-cost and flexible to modifications. Vision-based tactile sensors usually consist of a piece of elastomer, a camera, and a lighting system. When externally in contact with an object, the sensor captures the deformation of the elastomer by a camera and infers characteristics, such as the shape of the contact, and the shear and torsional forces.

While tactile sensors based on resistance, capacitance, and piezoelectricity [28, 158, 26] can be great options for regular linkage-based robotic hands, they are not suitable to be integrated with continuously rolling mechanisms. Because their design involves deploying wires/cables from the electronics to the sensing area, the wires/cables would inevitably get tangled due to the continuous rotation. Vision-based tactile sensors, in comparison, provide great advantages by allowing the tactile signal to be transferred through light, eliminating the mechanical coupling between the sensing area and the electronics making it an ideal choice for continuously rolling mechanisms.

There have been previous works [122, 19] that integrate vision-based tactile sensors into passive rollers for inspection tasks. It was demonstrated that rolling action greatly improves the efficiency of inspection especially when scanning large areas. However, since the rollers in these works are passive, they rely on the motions of robotic arm navigate through the inspection areas.

In this work, we integrate a category of vision-based tactile sensors - known as GelSight sensors [163] - into actively-driven rollers, which not only enables better capabilities for in-hand manipulation through rolling contact, but also has the ad-

vantage of efficiently inspecting the geometric properties of the grasped object during manipulation. In addition to the shear forces and 2D contact geometry, GelSight sensors can also provide high-resolution 3D contact geometry by applying photometric stereo [163]. The 3D information can be further processed and be used for normal forces estimation, pose estimation, and surface reconstruction. We design the sensor so that it can fit into a compact form-factor of the actively driven rollers while preserving the high-resolution 3D contact geometry.

4.3 Method

4.3.1 Sensor and Hand Design

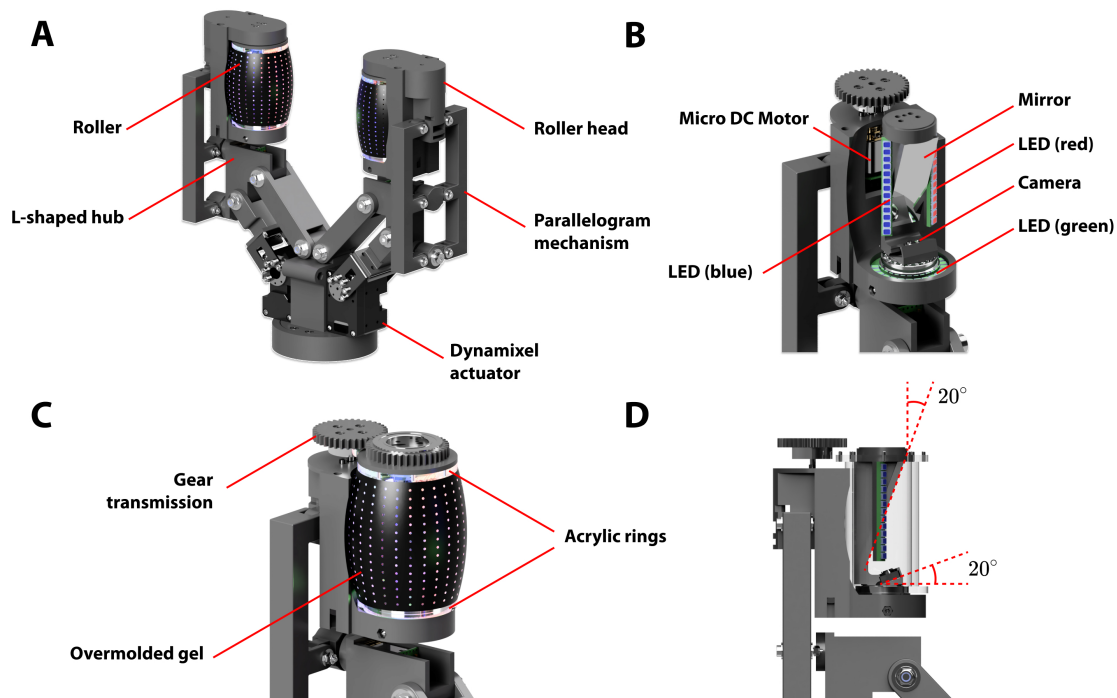


Figure 4-1: **CAD renderings of the mechanical design of Roller Grasper V4.** (A) Fully assembled hand. (B) Optical components required for the GelSight sensor located inside the roller. These components are not rotating with the roller. The non-rotating structure that houses the optical components is called stator. In comparison, the rotating part is called the rotor. (C) The roller (the unbounded rotating mechanism) (D) Arrangement of the camera and mirror inside the roller.

The TERG is a two-fingered grasper with each finger consisting of three actuated DoF. The design of the grasper is shown in Fig. 4-1. The base DoF is driven by a Robotis Dynamixel XM430-X350 actuator through a four-bar linkage mechanism. The mechanism allows the rollers to stay in parallel planes during manipulation and enables up to 160mm opening between the rollers. A micro DC motor embedded in the L-shaped hub controls the second DoF and is capable of pivoting the roller head up to $\pm 90^\circ$ through a five-bar parallelogram mechanism. The mechanism was improved from the parallelogram mechanism in the previous work [160] to allow for a greater range of motion. Another micro DC motor is embedded at the back of the roller head to drive the roller through spur gears. Unlike the previous generations of the Roller Grasper, the motor driving the roller is located outside the roller to make space for the optical components required for the tactile sensor.

One of the most important considerations for vision-based tactile sensors is designing for clear optical paths. Specifically, the space between the light source, the sensing area and the camera all need to be optically clear. As a result, the mechanical structure of the roller consists of a clear acrylic tube glued with two clear acrylic rings on both ends, allowing unobstructed light passage from the light source. A 3D-printed gear is attached to one of the acrylic rings. Eight dowel pins arranged around the perimeter are inserted into both the gear and the acrylic ring to provide sufficient torque transmission. The clear gel is molded directly over the acrylic tube for the best optical transparency.

The roller rotates around the stator of the roller head; both ends of the stator are rigidly connected to the rest of the roller head. As shown in Fig. 4-1D, a Raspberry Pi camera is located at the bottom of the stator. The camera is oriented 20° from its horizontal mounting surface and streams images from the sensing area through a mirror oriented 20° from the rolling axis. This optical design accommodates for the focal length of the camera inside the relatively narrow space inside the roller while maximizing the utilization of the camera’s field of view (FOV).

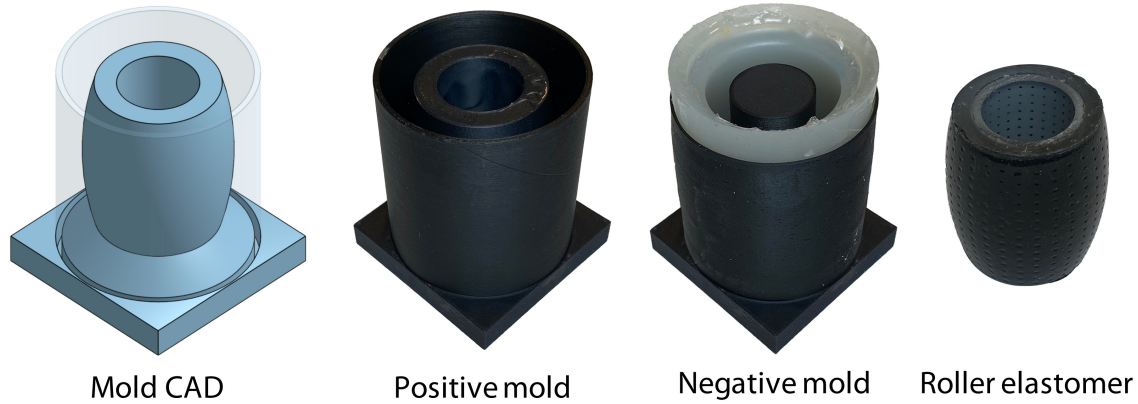


Figure 4-2: **Mold for the seamless roller elastomer.** From *Left to Right*: CAD model of the mold; 3D printed positive mold, with surface smoothed; rubber negative mold; seamless elastomer covering around clear acrylic tube.

4.3.2 Sensor fabrication

Lighting The lighting system was designed to have accurate 3D reconstruction based on photometric stereo, while fitting into the compact form-factor of the roller. To satisfy the requirements, we modified the design of the lighting system from the GelSight Wedge sensor [140] to be suitable to illuminate the curved roller surface.

As shown in Fig. 4-1, the clear acrylic tube located at the center of the roller provides mechanical support for the rotor while allowing light to shine through. A camera is mounted at the bottom of the stator, and captures the sensing area through a mirror. Two LED bars (one blue and one red, respectively) are located near either vertical edge of the mirror to provide directional light from two different directions toward the sensing area. A green LED ring attached below the roller shines light through the bottom acrylic plate to provide the third color component essential for 3D reconstruction.

In order for the lighting system to not be disrupted by the rotation of rollers, a clear acrylic plate was glued at the bottom end of the acrylic tube with clear UV resin (Ultraviolet Curing Epoxy Resin, Limino). The top end is constructed similarly for aesthetic purposes. The clear UV resin fills in the gap between the acrylic tube and plates, making the interface between the two surfaces optically clear for the light from the LED ring at the bottom to travel through.

Camera We used a Raspberry Pi camera with a 120° FOV, allowing us to obtain a relatively large sensing area while fitting the camera inside the tight interior of the roller. The camera was customized with a 200 mm long flex cable, so the bulky connector can be located outside the roller. We streamed the video at 30Hz through `mjpg_streamer` to the Raspberry Pi, with a 640x480 resolution. The images were then transmitted from the Raspberry Pi to a PC to be used for higher-level tasks.

Elastomer We designed and fabricated the seamless elastomer to obtain continuous tactile signals during rolling. In comparison, another fabrication technique is to cast a piece of flat elastomer to be wrapped around the rotor core[122, 19], which would be less durable and result in discontinuous sensing signals at the seam.

Fig. 4-2 shows the sequence of the elastomer fabrication. We first 3D printed the positive mold and smoothed the curved surface with a layer of coating (XTC-3D, Smooth-On, Inc.). A stretchable negative mold was then cast using the translucent silicone (Ecoflex, Smooth-On, Inc.). Next, the clear silicone (XP-565, Silicones, Inc.) on the roller was cast together with the acrylic tube. We applied a layer of primer (DOWSIL PR-1200, RTV Prime Coat, DOW) on the outside of the acrylic tube before casting to increase the acrylic-to-elastomer bonding. Finally, we sprayed a layer of opaque gray (Lambertian) silicone inks (Print-On Silicone Ink, Raw Material Suppliers) onto the surface of the roller.

Markers To provide information of shear and torsional forces, we added multiple arrays of markers on the surface of the roller. The markers were directly lasercut around the curved surface of the roller using a CO_2 laser cutter with a rotary attachment. The laser cutter etched away the gray coating at each pre-defined marker location over the entire surface of the roller, leaving only the transparent silicone exposed. Finally, we applied a layer of black silicone ink (Print-On Silicone Ink, Raw Material Suppliers) on the surface of the roller. The resulting roller presents black markers with a gray background in the camera view.

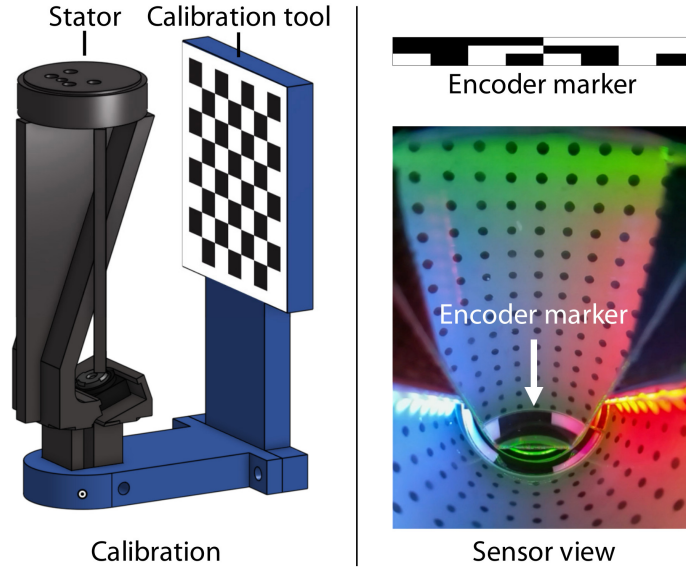


Figure 4-3: **Camera calibration and encoder marker.** *Left:* A 6x7 Checkerboard mounted on a calibration tool to get the camera intrinsic matrix, and the corresponding extrinsic matrix in the roller frame; *Right:* The pattern of the encoder marker to provide precise position encoding, and the corresponding image from the sensor view.

4.3.3 Tactile Signal Processing

This section discusses the signal processing techniques for the raw tactile signal. It also addresses the challenges created by the continuous rolling and convex sensing surface, and the corresponding solutions that we proposed.

Encoding In order to achieve 3D reconstruction and marker tracking, the signal processing algorithms require each image to be compared with a reference image taken in the absence of contact [163]. Unlike GelSight sensors with the conventional form-factor, our sensing area expands the entire perimeter of the roller and thus multiple reference images in correspondence to different roller positions need to be taken in order to properly process the sensor signal. This requires the algorithm to find the correct reference image. However, due to backlash in the transmission and latency between the actuator and camera, the roller motor encoder cannot be used to correspond a given image to its designated reference. Therefore, we attached an encoder inside the camera FOV, as shown in Fig. 4-3, in order to match a given image with its reference. The encoder designed with this method can achieve pixel-

level precision.

During the calibration process, the roller slowly rotates at a constant speed, allowing the camera to record reference images along with the encoder images in order to construct a lookup table for each frame. During manipulation, we extract the encoder portion of the image and find the L2 distance between the current encoder image and references from the lookup table to determine the corresponding reference image. Finding the correct reference image is a crucial early step toward the successful processing of tactile signals.

Surface Projection Camera matrices are used to calculate the correspondence between the points on the sensor surface in 3D and the 2D camera image pixels. The transformation [129] can be represented as:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R|t] \begin{bmatrix} X_6 \\ Y_6 \\ Z_6 \end{bmatrix} \quad (4.1)$$

where $(u, v)^T$ represents the image coordinates of the sensor input; λ is a scale factor; K is the camera intrinsic matrix; $[R|t]$ is the camera extrinsic matrix, with rotation R and translation t ; $(X_6, Y_6, Z_6)^T$ represents the 3D coordinates in the sensor frame, shown as Frame A_6/B_6 in Fig. 4-11.

The camera was calibrated using a 7x6 checkerboard. The camera, along with the mirror, was first mounted to the 3D printed housing and calibrated before the stator was assembled with the rest of the roller head. During camera calibration, multiple sensor images were collected with different checkerboard poses, which were later used for providing the camera intrinsic matrix K . The extrinsic matrix $[R|t]$ was derived by taking the image of the checkerboard and using its known position with respect to frame A_6/B_6 when it is rigidly mounted on the stator, as shown in Fig. 4-3. We applied OpenCV *calibrateCamera* [14] to the image pixels and their corresponding 3D positions to get the intrinsic matrix K and the extrinsic matrix $[R|t]$.

3D Reconstruction The 3D positions of the points on the convex sensing area can be projected from the Cartesian space to the 2D camera image space using the

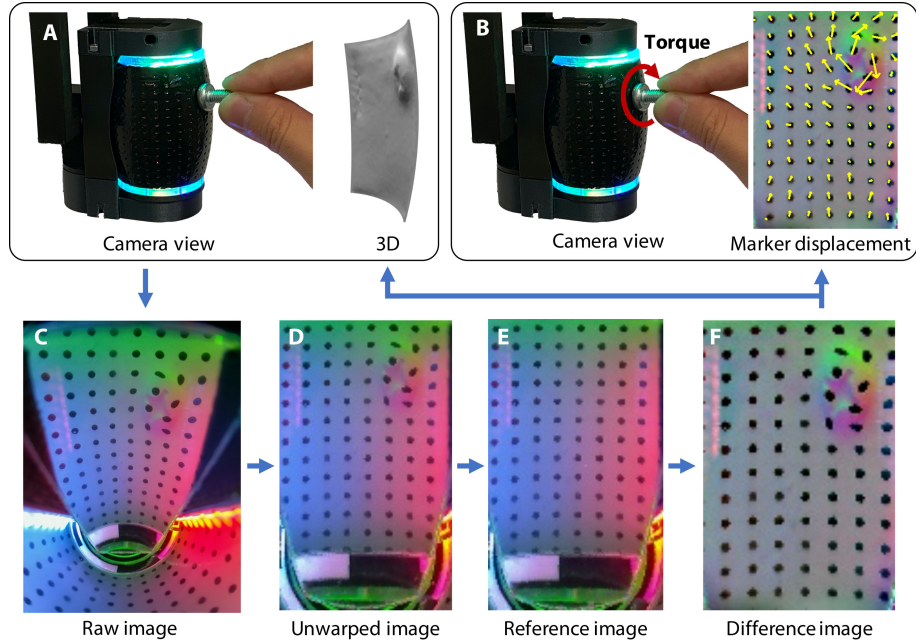


Figure 4-4: **3D reconstruction and marker tracking for the roller sensor.** (A) The camera view shows a screw head pressing on the roller sensor, and the 3D shows the estimated 3D reconstruction. (B) The camera view demonstrates the torque exerted on the roller sensor, and the marker displacement visualizes the magnified motion of the markers captured from the sensor. (C) The camera inside the roller sensor captures the raw image, and the sensing area is captured in the mirror. (D) The raw image is unwarped into a rectangular image. (E) The reference image is extracted with the encoder marker from the unwarped image. (F) The difference image is calculated between the unwarped image (after contact) and the reference image (before contact). It is further processed to get the 3D reconstruction, and marker displacement.

camera matrices. Because the geometry of the roller is known, this projection also allows us to trace the 3D position of a point given its 2D coordinate in the image. This mapping of the points on the sensing area between their 3D positions and 2D image pixels are saved for 3D reconstruction when an external object is in contact with the roller.

As shown in Fig. 4-4, when an object is in contact the roller the elastomer on the roller is deformed, creating a shaded image that is recorded by the camera. After unwarping the image into a rectangular shape (with the same pixel density along its horizontal and vertical axes), we applied photometric stereo to create a depth image: each pixel on the depth image will have a corresponding depth value, indicating the

offset from its position on the undeformed roller surface. We apply this depth image on top of the mapping described previously to reconstruct the 3D geometry of the contacted object. This is accomplished by subtracting the offset of each pixel in the depth image from its corresponding 3D position along the surface normal direction.

The photometric stereo technique used in this work is based on the previous work in 3D reconstruction using a planar elastomer. Specifically, we first transformed the shaded image into surface normals, and then applied the fast Poisson solver [33] for integration to produce the depth image. Further details of this method can be found in [163].

Marker Tracking The shear force estimation can be obtained by motion analysis, i.e., analyzing the marker displacement on the sensor in comparison with the reference images. During operation, markers are constantly disappearing and appearing from the boundaries of the sensor image due to the rotation of the roller, making the calculation of the marker displacement field difficult. A sensor image might even have a different number of markers compared to its reference image because certain markers are located at the very edge of the sensing area. In such situations, techniques using marker tracking with nearest temporal matching [163, 150] or optical flow [118, 170] tend to generate erroneous results. We adopted Random Optimization to reliably track marker displacement during rolling. We randomly sampled possible solutions to maximize the marker flow smoothness while minimizing the mismatching penalty. Specifically, the marker flow smoothness describes the phenomenon that nearby points move with similar velocities [62]. The flow smoothness of each marker is the difference between its displacement and the average displacement of its surrounding markers. The total flow smoothness of the sensing area is the sum of the flow smoothness of individual markers. The mismatching penalty is designed to handle the situation when a sensor image does not have the same number of markers compared to the reference image, which adds a penalty for each marker that does not have a corresponding marker in the reference frame.

One of the most important steps of computing the marker displacement is to match each marker in the sensor image to a corresponding marker in the reference

image. A possible option is to use the greedy approach, which matches each marker in the sensor image to the closest marker in the reference image. While it might work in certain situations, it would fail completely when the shear force is large enough to displace a marker for over half of the marker spacing, because it will recognize its neighboring marker in the reference image as its correspondence. In contrast, the Random Optimization uses a weighted sampling technique, where markers (in the sensor image and the reference image, respectively) with closer distances will have a higher probability to be matched. Because the sensor image and the reference image can have different numbers of markers, our algorithm does allow markers to have no matching. In addition, each marker in the sensor image is only allowed to be matched to up to one unique corresponding marker in the reference image.

Our method is also designed for real-time signal processing. While an exhaustive search can give us similar final results, it is also computationally expensive, which is not suitable for online operations. To further speed up our algorithm, we implemented the code in C++ with Python bindings. We sampled 200 possible solutions for each frame, allowing the algorithm to achieve real-time marker tracking at the frequency of 30 Hz.

4.3.4 Control methods for In-Hand Manipulation

We developed a series of demonstrations for TERG to demonstrate its capabilities. While these demonstrations required various high-level control methods, the low-level joint space control method is consistent across all of them. The base joints used current-limited position control to ensure that the object is being grasped securely without excessive internal force. Position control is used to drive the pivot angle between $\pm 90^\circ$. Smooth rolling motion is achieved through velocity control of the rollers. A summary of the control strategies for different demonstrations is shown in Table 4.1 Some of the demonstrations were carried out both with and without the sensor feedback in order to highlight the benefits of tactile sensing.

One of the most direct ways of using tactile information is to extract the contact location of the object to use as the control input. For well-defined simple tasks,

Table 4.1: Control Strategies for Manipulation Demonstrations

Object	Cylindrical	Planar	Spherical	Cable		Card
Input	u	v	x_G, y_G	v	f_u	f_u
Target	0	0	Tr_G	0	f_{target}	heuristic rules
Control	ω_r	θ_p	ω_r	θ_p	ω_r	ω_r

control can be done directly in the image space where we control the joint output based on the contact location in the image space. The contact location is obtained by locating the coordinate within the contact area that has the largest indentation.

Cylindrical object rotation For example, in the manipulation of a cylindrical object (Fig. 4-5), the controller adjusts the rolling speed of the rollers to keep the contact location of the pen and roller at the center of the processed image. We found the controller is robust even in the presence of external disturbances as the pen was kept within grasp our experiments. In addition, we can extract the primary and secondary principal axes of the contact area using principal coordinate analysis [120]. For an object with a relatively consistent contact shape, the primary and secondary principal axes indicate the orientation of the contact geometry, and subsequently the pose of the grasped object. In this particular example, the principal axis indicates the long axis of the pen.

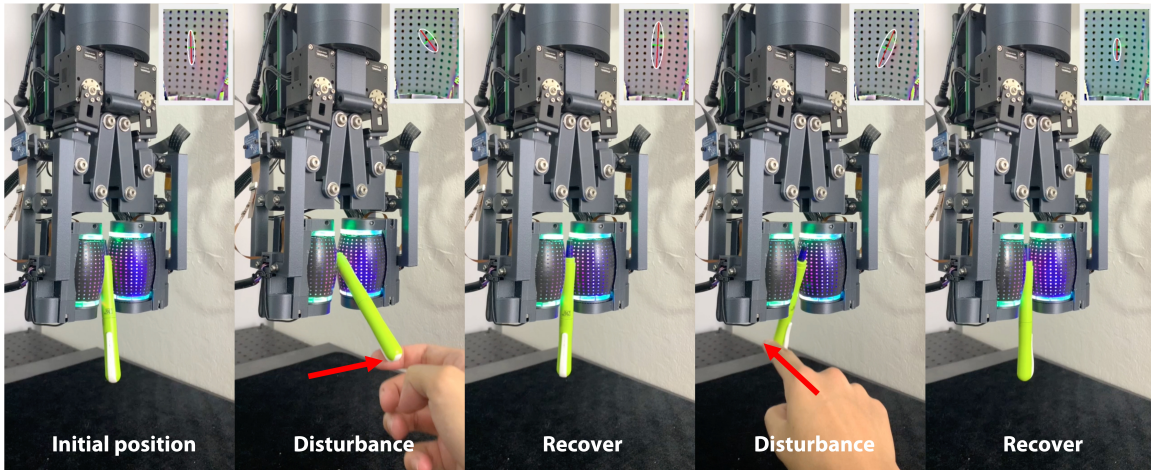


Figure 4-5: In-hand manipulation - cylindrical object.

Planar object reorientation The planar object reorientation (Fig. 4-6) also uses the contact coordinate in the image space as feedback and uses the pivot joint

angle as a control output. The planar objects used in this demonstration are 3D printed with varying radii of curvature to demonstrate that the control method can adapt to complex and unknown 2D geometries.

Both of the previous two examples were also run in open-loop without the sensor feedback, and the grasped objects were dropped shortly after the beginning of the manipulation. Without the tactile information, there is no way for the high-level controller to deduce the object status.

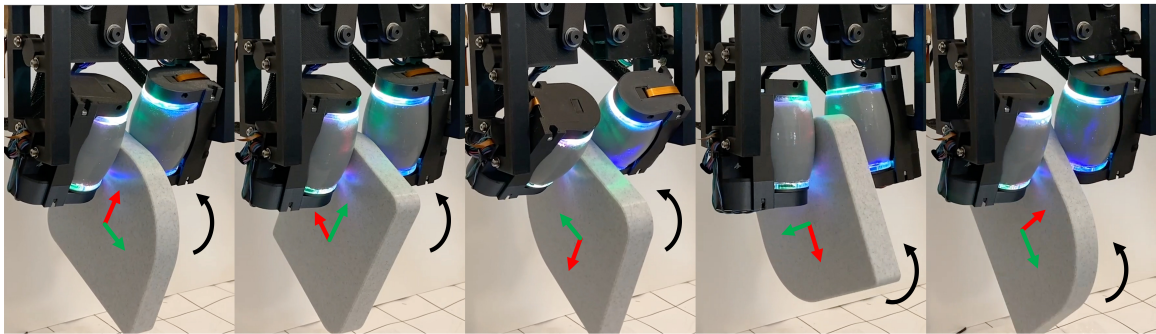


Figure 4-6: **In-hand manipulation - planar object.**

Spherical object screw motion and trajectory following Unlike the previous two demonstrations with a fixed control target, the spherical object manipulation task (Fig. 4-7) attempts to move the object along pre-defined trajectories in the operational space. With a known sphere diameter, we can compute the position of the object through the forward kinematics, which is then used to close the high-level trajectory following control loop. In this example, the object goes through screw motions with the translational and rotational motions coupled according to the relative pivot angles of the two rollers. We also specifically picked a transparent object to demonstrate the benefit of using tactile sensors for objects that are challenging to track using other techniques such as vision-based tracking.

Note that in this demonstration, the acrylic marble achieved screw motions, which are difficult or impossible to perform with traditional robot hands. TERG easily achieves screw motions by setting the rollers in opposite orientations, forming a cross. Changing angle between the rollers enables setting the screw pitch from zero to infinity, and anywhere in between.



Figure 4-7: **In-hand manipulation - spherical object.**

In addition to the in-hand manipulation demonstrations that only use depth information from the tactile sensor, we can combine the depth and shear information to achieve a more comprehensive manipulation demonstration, as shown in the two examples below.

Cable tracing Unlike rigid body objects, cables can withstand substantial tensile load but can buckle under modest compressive loads. This makes a cable a very difficult object for robot manipulation. Even when humans try to trace along a cable, a typical practice is to use our fingers to grasp onto a location on the cable near where it is anchored and then slide along the cable. Such motion can only be achieved while sliding away from where the cable is anchored in order to maintain the cable tension, rather than towards the anchored location which will apply a compressive load due to friction. The combination of the rolling motion and the ability to compute shear force applied by the cable makes it possible for the rollers to trace along a cable both toward and away from the anchor point while keeping the cable in tension. On the other hand, the contact location is also computed in this case to make sure that the cable does not get dropped due to gravity during tracing. This is achieved by adjusting the pivot direction in response to the changing contact locations between the cable and the roller. We tested two open-loop cases for this demonstration. The first case does not

use any sensor feedback, which results in the cable being dropped almost immediately; without monitoring the cable contact location, the rollers cannot adjust their pivot accordingly and the cable is lost. The second case tracks the contact location but ignores the shear information. The rollers are able to move along the cable for longer, however, the cable eventually became slack and was no longer traversable.

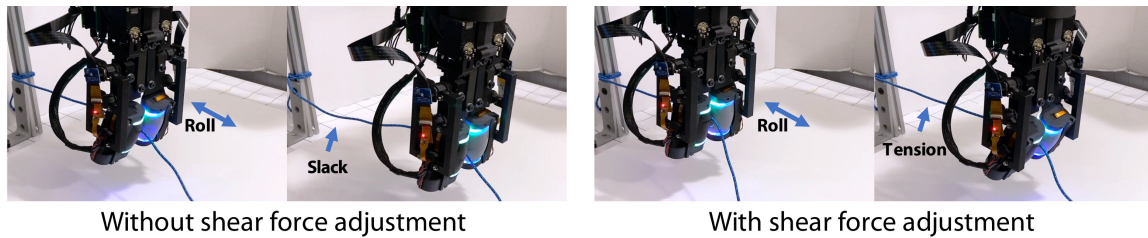


Figure 4-8: **Cable tracing.** The roller grasper can reactively roll along cables back and forth without losing the cables. The tactile sensor provides the contact location of the cable and estimated shear forces in real time. The contact location is used to modulate the pivoting angle to compensate for the cable gravity during rolling. The shear force is used to keep the tension during rolling. *Left:* Without the shear force adjustment, the cable can be accumulate slack over time. *Right:* With the shear force adjustment, the roller can consistently maintain the tension of the cable over time.

Card picking Another way of using the tactile information is to capture the transient events, which is also frequently done by humans during our daily activities. One of the difficulties in interfacing with thin objects, such as paper or playing cards, is detecting the number of pieces within the hand. Because of their extreme aspect ratios, it is very likely for multiple pieces of paper or cards to stick together when they get picked up. To distinguish between multi-card and single card situations we can actuate one side of a given card and monitor the amount of shear force observed on the tactile sensor. In the case when multiple cards are within the hand, the relative motion of any two cards will reduce the amount of shear force created on the roller. However, if the shear force suddenly increases, this indicates that only one card is left in grasp. This is only one specific example demonstrating the transient property of the shear information. In practice, there are a variety of situations where this methodology can be used [137, 23, 77], especially in cases where the state of the hand-object configuration experiences a sudden change.

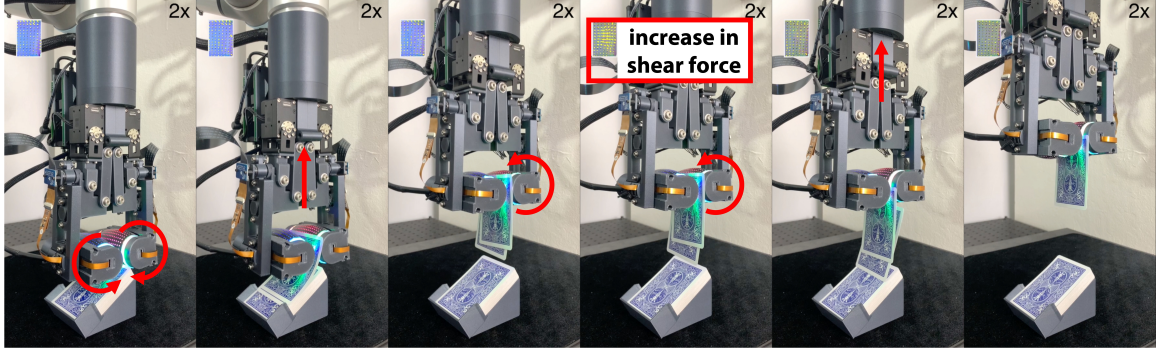


Figure 4-9: **Card picking.** We distinguish whether there is only one card picked up by actuating one roller and monitoring the change in shear force. When the number of card within hand is reduced to one, there will be a increase in shear force detected by the tactile sensor.

4.4 Results

The advantages of combining the Roller Grasper and the GelSight sensor are two-fold. First, incorporating tactile sensing greatly improves the in-hand manipulation capabilities of the Roller Grasper by enabling the grasper to detect local contact information between the rollers and the grasped object. Second, the steerable rollers enable the tactile sensor to easily scan potentially large and complex surfaces, leading to efficient and accurate 3D reconstructions. Our robotic system includes the TERG, a Universal Robots UR5e robot arm, and a computer, as shown in Fig. 4-10.

4.4.1 In-Hand Manipulation with Tactile Sensing

In terms of kinematics, the two-finger design with six total actuated DoFs for the TERG is a significant simplification compared to the previous Roller Graspers (three-finger Roller Grasper V1/V2 and four-fingers Roller Grasper V3). This simplification was made possible largely because of its tactile sensing capabilities, allowing it to actively manipulate the grasped object based on the contact information without the need for the extra grasp stability provided by the grasper’s redundancies present in the previous versions. Even with the vastly reduced DoFs, TERG is still capable of translating or rotating the grasped object in each of the X_O , Y_O and Z_O directions (defined in Fig. 4-11A). The manipulation directions that the grasper can impart

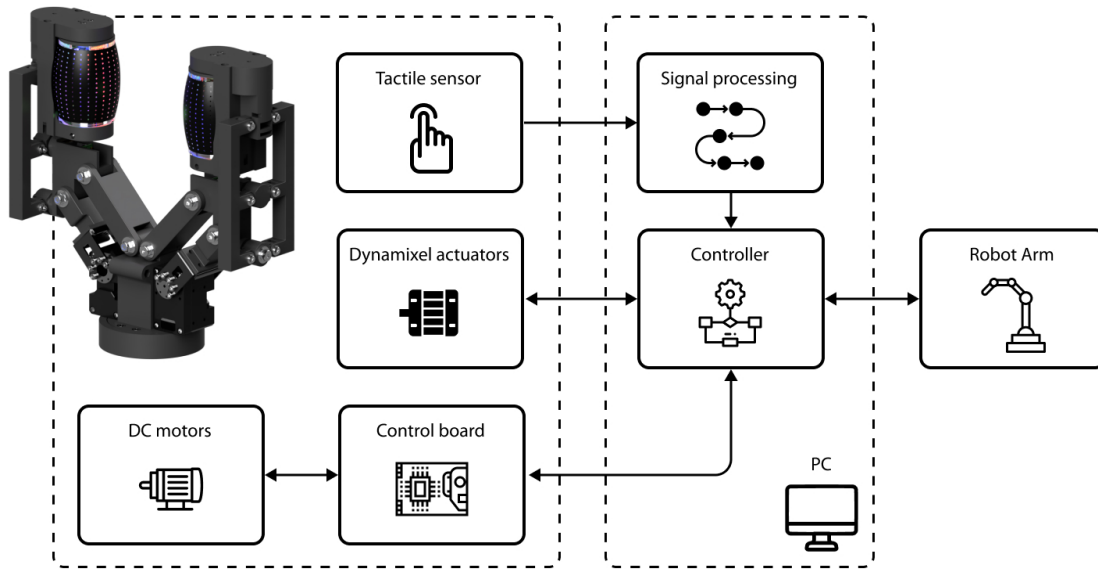


Figure 4-10: **System Diagram**

on the grasped object are presented in Fig. 4-11. The combination of the different manipulation directions allows the grasped object to be manipulated between a wide variety of initial and target poses.

The tactile sensor provides both depth and shear information for an object in contact with the rollers. The raw sensor data can be further processed to extract higher-level information suited for in-hand manipulation. In the acrylic ball manipulation demonstration (Fig. 4-7), we extracted the real-time contact center of the ball using the depth information, which was used in the forward kinematics for an operational space trajectory following. In the cable manipulation demonstration, the contact center was used for tracking the contact location of the cable to adjust the pivot joint in order to prevent the cable from slipping out due to gravity. At the same time, we adjust the rolling speed to maintain the cable’s tension by tracking the shear force applied on the roller by the cable.

The tactile sensor also helps mitigate certain hardware limitations. For example, the transmission ratio between the the roller and the actuator is relatively high in order to achieve ample torque from the micro DC motor selected for the compact form factor, which makes the roller non-backdrivable. The tactile sensor enables a

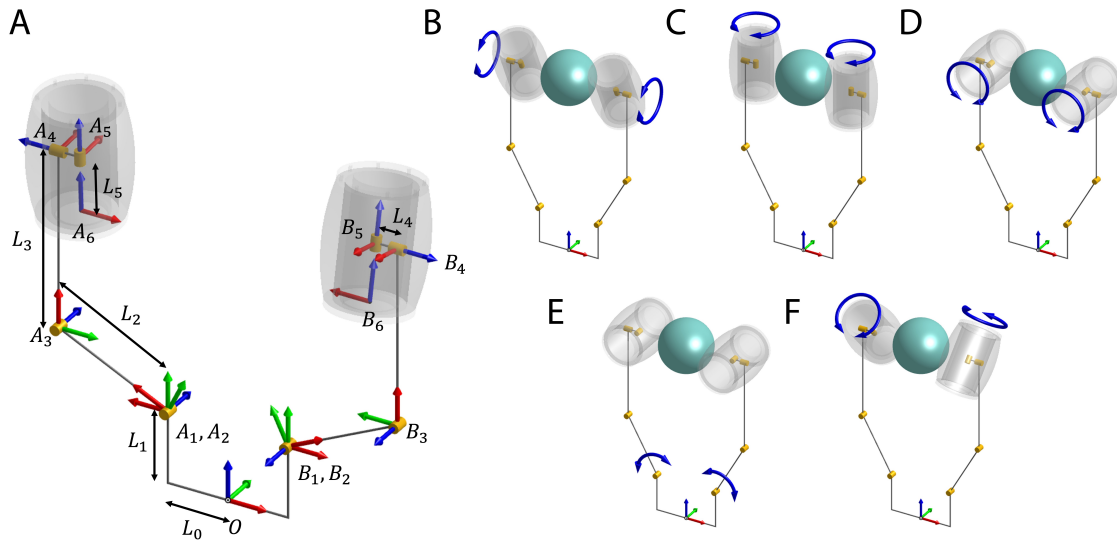


Figure 4-11: **3D kinematics and roller configurations.** (A) Roller Grasper V4 frame definitions. The Two fingers are represented by letters A and B , respectively. Frame O is the hand fixed frame located at the base of the hand. The numerical subscripts represent frames attached at different locations of the hand. Frames 1-5 are attached at different joints while Frame 6 is at the bottom of the roller used as the reference frame for sensor image. The X , Y and Z axes are represented in red, green and blue colors, respectively. Frame O is the world frame with which we reference the manipulation directions. (B) Object rotation in X_O . (C) Object rotation in Z_O or object translation in Y_O , depending on the rolling directions of the two rollers. (D) Object rotation in Y_O or object translation in Z_O , depending on the rolling directions of the two rollers. (Any rotation or translation in directions within $Y_O - Z_O$ plane are possible with different pivot positions) (E) Object translation in X_O . (F) Object screw motion (coupled rotation and translation)

force control along the shear direction, which opens up additional abilities for object manipulation and safe interaction. The rollers could either not react to shear force (As shown in Fig. 4-12A), taking advantage of the friction of the transmission for secure grasping and in-hand manipulation, or actively adjust for the speed based on the shear information (As shown in Fig. 4-12BC), allowing for a compliant manipulation or safe external interactions.

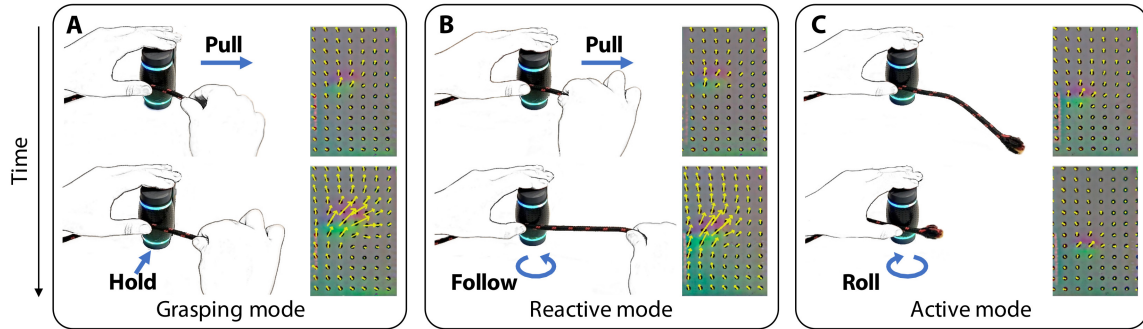


Figure 4-12: **Roller actuation modes.** (A) The roller can hold the current position to grasp the cable, resisting external forces. The marker displacement from the sensor images indicate the exerted external forces. (B) The roller can reactively roll along the cable, following the external forces. (C) The roller can actively roll along the cable, without external forces.

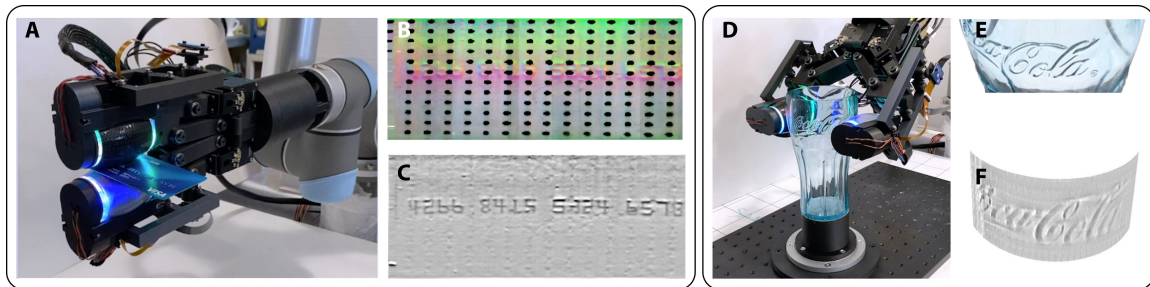


Figure 4-13: **Surface scanning.** (A) Rolling along a credit card. (B) Stacked tactile images in the time sequence, showing the embossed numbers on the credit card. (C) Processed tactile image with interpolation at the marker region and sharpening filters for better visualization. (D) Rolling along a transparent cup. (E) The embossed characters on the cup. (F) Scanned tactile images stitched in 3D spaces.

4.4.2 Efficient object/image reconstruction using Steerable rollers

The rolling action combined with tactile sensing results in efficient surface inspection or reconstruction, compared with previous methods where researchers needed to apply a long and slow sequence of discrete touches when inspecting surface roughness after sanding[5], detect defects on objects [40, 72], and reconstruct 3D shapes[12, 141, 124].

We demonstrated this ability through the reconstruction of surface geometries for both a credit card (2D) and a transparent cup (3D). In the 2D demonstration, the grasper carefully guided the credit card in between its rollers while scanning the surface textures of the card, as shown in Fig. 4-13. The tactile images were then

stacked in the time sequence to recover the credit card numbers. We also show the raw stacked images in comparison with the processed images to better visualize the effects of filters and the interpolation required to fill the areas occluded by the black markers.

The 3D demonstration reconstructed a transparent cup with an embossed logo. The cup was mounted on a turntable to enable pure rotation around Z_O . The surface of the cup was inspected through a series of motions. The rollers held onto the opposing sides of the cup near its opening, and traversed from the lip to its base; the rollers then tilted a small angle and rolled upwards until they reached the opening of the cup at adjacent positions to where they started. The slightly tilted angles of the rollers resulted in a screw motion of the cup: every time when the rollers rolled down and up, the cup was rotated by a small angle. The sequence would repeat until the scanning of the entire surface was completed. To better align the multiple scanned images, we applied cross-correlation between them to match their correspondence. The resulting images are presented in point clouds, as shown in Fig. 4-13. We specifically picked a transparent cup to show the capabilities of this technique, which can be challenging to accomplish when using color or depth cameras.

4.4.3 Contributions

This work presents the design of a Roller Grasper that has steerable active rollers at fingertips integrated with high-resolution tactile sensors. We developed algorithms to process tactile signals in order to provide real-time feedback for in-hand manipulation and object reconstruction. Incorporation of tactile sensing and continuously rotating mechanisms is a nontrivial problem, but through clever mechanical and algorithmic design we were able to overcome these challenges. To the best of the author's knowledge, this is the first time a robot grasper has demonstrated the ability to perform robust in-hand manipulation for various objects through tactile-guided rolling contact, even with unknown dynamics or external disturbances. We also demonstrated its unique capability to efficiently retrieve object geometries (both in 2D and 3D) during manipulation. This was only made possible with the combination of actively

driven rolling contact and high-resolution tactile information.

In summary, we presented the abilities of the Roller Grasper for in-hand manipulation and object reconstruction as well as its potential to complete complex perception and manipulation tasks in various real-world robotic settings. We hope this work would push the boundaries in both robotic manipulation and tactile sensing.

4.4.4 Limitations and future works

While we have demonstrated the incredible abilities of TERG, there are different aspects of this work that can be further explored.

In terms of the design of the grasper, although rollers on TERG have a convex curvature, a spherical shape roller would ultimately provide better grasp stability for objects with complex shapes. Another limitation of the current design is that the size of the sensing area is restricted by the 90° camera field of view, which could be increased by using a camera with larger FOV, i.e., a fish-eye camera.

Our demonstrations presented feedback control methods using only tactile information, however, inclusion of additional sensing modalities could provide both global object information as well as local contact information.

TERG’s control pipeline could be augmented through integrating its object geometry reconstruction and in-hand manipulation abilities: while TERG can manipulate objects with unknown geometry and dynamics, the geometry of the object reconstructed during manipulation can further be used to improve the manipulation results.

Chapter Acknowledgement

This chapter was a joint work with Shenli Yuan, Radhen Patel, Megha Tippur, Connor Yako, Edward Adelson, and Kenneth Salisbury. Toyota Research Institute (TRI), and the Office of Naval Research (ONR) [N00014-18-1-2815] provided funds to support this work. Shenli Yuan was supported, in part, by the Stanford Interdisciplinary Graduate Fellowship. The authors would also like to thank Sandra Liu for setting up

the laser cutter with the rotary attachment.

Chapter 5

Conclusion

This thesis has explored different directions to apply touch to improve robot dexterity in manipulation tasks more interactively. In this chapter, we will summarize what we have learned, and propose several future directions.

5.1 Summary of Contributions

We started with cable manipulation, introducing tactile-reactive control for the sliding motion. We used real-time tactile feedback to accomplish the task of following a dangling cable. Touch provided the pose of the cable in the grip, and the friction forces during cable sliding. Because the cable is deformable and moves in the free space, the robot can interactively perceive and change the cable state, with the help of touch. To make the control effective but simple, we decoupled controller into two tactile-based controllers: 1) Cable grip controller, where a PD controller combined with a leaky integrator regulates the gripping force to maintain the frictional sliding forces close to a suitable value; and 2) Cable pose controller, where an LQR controller based on a learned linear model of the cable sliding dynamics keeps the cable centered and aligned on the fingertips to prevent the cable from falling from the grip. This controlled-sliding motion is possible by a reactive gripper fitted with GelSight-based high-resolution tactile sensors. The robot can follow one meter of cable in random configurations within 2-3 hand regrasps, adapting to cables of different materials and

thicknesses. The tactile-reactive behavior made it robust to external disturbances.

We can further improve the dexterity of the robots with dynamic movements. Such tasks are susceptible to variations in the physical properties of the manipulated objects. In SwingBot, we presented a robot system that can learn the physical features of a held object through tactile exploration, and apply the learned features to the dynamic swing-up manipulation task. Because the dynamic movement is fast, the robot gathered tactile exploration data during various interactions in this task and planned for the action to perform the dynamic manipulation. Two exploration actions (tilting and shaking) provided the tactile information used to create a physical feature embedding space. With this embedding, SwingBot can predict the swing angle achieved by a robot performing dynamic swing-up manipulations on a previously unseen object. Using these predictions, the robot can search for the optimal control parameters for a desired swing-up angle. We showed that, with the learned physical features, our end-to-end self-supervised learning pipeline could improve the accuracy of swinging up unseen objects substantially. We also show that objects with similar dynamics are closer to each other on the embedding space and that the embedding can be disentangled into values of specific physical properties.

Besides using dynamic movement from the robot arm, another direction to improve robot manipulation is to increase in-hand tactile dexterity. With the Tactile-Enabled Roller Grasper (TERG), we explored novel ways to manipulate objects by combining active rolling and tactile sensing. The Roller Grasper designs allow the grasped objects to be rotated or translated within hand while maintaining stable grasps. In addition, its novel capabilities were greatly enhanced by adding tactile sensing. Such sensing provided information about an object’s local shape and texture, and was used in feedback for controlled manipulation. The capabilities of the TERG were demonstrated through a series of comprehensive in-hand manipulation and object reconstruction tasks. Specifically, we showed that the tactile feedback allowed the grasped object to be manipulated stably and continuously, resisting external disturbances, in contrast to the situations without feedback. We believe that the combination of active surface and tactile sensing on the robot end-effector opens

up a whole new range of possibilities for robot in-hand manipulation.

In summary, this thesis introduced various ways to improve robot dexterity through interactive touch. Compared to vision, touch provides unique information about the contact geometry (in-hand pose of cables/pens/planar objects, etc.), and contact forces (forces along sliding, rolling, and exploratory procedures, etc.). This information is crucial for handling contact, but inherently challenging to perceive from vision. I believe touch can really shine for robot manipulation and hope this thesis provided some directions to push forward the boundaries of touch and manipulation research.

5.2 Future Work

Sensor and Hand Design There are many aspects to improve in terms of hardware design. For example, for tactile sensor design, it is desired to 1) provide dense contact geometry and forces for fine/contact-rich manipulation 2) be compact enough to fit into various robotic hands; 3) be multi-directional to sense the contact not only from the front but also from the side/back; 4) provide large coverage over the hand; 5) be multi-modal to sense richer information such as vibration, proximity, etc; and more. Based on applications, there will be different focuses and compromises to achieve the goal, which is exciting to keep exploring. For hand design, it can be viewed as a co-design of tactile sensor and hand. A dexterous hand without tactile sensing can be extremely difficult to control. Tactile sensors without more degrees of freedom can provide limited tactile dexterity. It would be simultaneously optimized design for both sensor and hand to achieve more dexterous manipulation tasks.

Multi-modal Learning Humans use multi-modal sensory data all the time. Touch alone provides unique contact information, but touch is local and only provides signals after contact. With the complementary information of vision or sound, the robot can perceive the global context better. Robots can use different modalities for cross-modal learning of the common representation, and multi-modal learning of the complimentary representation. The inherently self-supervision between different modalities can make learning more efficient. In addition, combining the unique aspect

of each modality can make the policy more robust.

Scaling up Robot Learning Large datasets contribute significantly to the success of machine learning models. Unfortunately, there are not many large touch datasets yet. With the emerging touch simulators and commercial touch sensors, high-quality large datasets become necessary and can benefit the community to thrive. Touch might be easier to transfer between different workspaces compared to vision, since touch has more structured signals and does not get influenced by the outside environments. However, because different tactile sensors can have different physical properties (hardness, illumination, marker density, etc.), calibration to the common representation (contact location, depth, dense forces field, etc.) can potentially lead to more efficient transfer.

Bibliography

- [1] Frank Abegg, Axel Remde, and Dominik Henrich. Force-and vision-based detection of contact state transitions. In *Robot manipulation of deformable objects*, pages 111–134. Springer, 2000.
- [2] Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in neural information processing systems*, pages 5074–5082, 2016.
- [3] Thamer Albahkali, Ranjan Mukherjee, and Tuhin Das. Swing-up control of the pendubot: an impulse–momentum approach. *IEEE Transactions on Robotics*, 25(4):975–982, 2009.
- [4] Alex Alspach, Kunimatsu Hashimoto, Naveen Kuppuswamy, and Russ Tedrake. Soft-bubble: A highly compliant dense geometry tactile sensor for robot manipulation. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 597–604. IEEE, 2019.
- [5] Alexander Amini, Jeffrey I Lipton, and Daniela Rus. Uncertainty aware texture classification and mapping using soft tactile sensors. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4249–4256. IEEE, 2020.
- [6] Shorya Awtar, Alexander H Slocum, and Edip Sevincer. Characteristics of beam-based flexure modules. *Journal of Mechanical Design*, 129(6):625–639, 2007.
- [7] Maria Bauza, Ferran Alet, YC Lin, T Lozano-Perez, L Kaelbling, P Isola, and A Rodriguez. Omnipush: accurate, diverse, real-world dataset of pushing dynamics with rgbd images. In *NeurIPS Physics Workshop*, 2018.
- [8] Maria Bauza, Oleguer Canal, and Alberto Rodriguez. Tactile mapping and localization from high-resolution tactile imprints. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3811–3817. IEEE, 2019.
- [9] Dmitry Berenson. Manipulation of deformable objects without modeling and simulating deformation. In *IEEE International Conference on Intelligent Robots and Systems*, 2013.

- [10] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. Discrete elastic rods. *ACM Transactions on Graphics*, 2008.
- [11] Walter G Bircher, Andrew S Morgan, Kaiyu Hang, and Aaron M Dollar. Energy gradient-based graphs for planning within-hand caging manipulation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2462–2467. IEEE, 2019.
- [12] Marten Björkman, Yasemin Bekiroglu, Virgile Högman, and Danica Kragic. Enhancing visual perception of shape through tactile glances. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3180–3186. IEEE, 2013.
- [13] Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017.
- [14] Gary Bradski. The opencv library. *Dr. Dobb’s Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000.
- [15] Rodney A Brooks. Elephants don’t play chess. *Robotics and autonomous systems*, 6(1-2):3–15, 1990.
- [16] Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H Adelson, and Sergey Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018.
- [17] Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H Adelson, and Sergey Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018.
- [18] Roberto Calandra, Andrew Owens, Manu Upadhyaya, Wenzhen Yuan, Justin Lin, Edward H Adelson, and Sergey Levine. The feeling of success: Does touch sensing help predict grasp outcomes? *arXiv preprint arXiv:1710.05512*, 2017.
- [19] Guanqun Cao, Jiaqi Jiang, Chen Lu, Daniel Fernandes Gomes, and Shan Luo. Touchroller: A rolling optical tactile sensor for rapid assessment of large surfaces. *arXiv preprint arXiv:2103.00595*, 2021.
- [20] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

- [21] Yevgen Chebotar, Karol Hausman, Zhe Su, Gaurav S Sukhatme, and Stefan Schaal. Self-supervised regrasping using spatio-temporal tactile features and reinforcement learning. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1960–1966. IEEE, 2016.
- [22] Ning Chen, Hong Zhang, and R Rink. Edge tracking using tactile servo. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 2, pages 84–89. IEEE, 1995.
- [23] Wei Chen, Heba Khamis, Ingvars Birznieks, Nathan F Lepora, and Stephen J Redmond. Tactile sensors for friction estimation and incipient slip detection—toward dexterous robotic manipulation: A review. *IEEE Sensors Journal*, 18(22):9049–9064, 2018.
- [24] Cheng Chi and Dmitry Berenson. Occlusion-robust deformable object tracking without physics simulation. In *Intelligent Robots and Systems (IROS), 2019 IEEE International Conference on*. IEEE, 2019.
- [25] Shadow Robot Company. Design of a dextrous hand for advanced clawar applications. CLAWAR, 2003.
- [26] Mark R Cutkosky and William Provancher. Force and tactile sensing. In *Springer Handbook of Robotics*, pages 717–736. Springer, 2016.
- [27] Nikhil Chavan Daffe, Alberto Rodriguez, Robert Paolini, Bowei Tang, Siddhartha S Srinivasa, Michael Erdmann, Matthew T Mason, Ivan Lundberg, Harald Staab, and Thomas Fuhlbrigge. Extrinsic dexterity: In-hand manipulation with external forces. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1578–1585. IEEE, 2014.
- [28] Ravinder S Dahiya, Giorgio Metta, Maurizio Valle, and Giulio Sandini. Tactile sensing—from humans to humanoids. *IEEE transactions on robotics*, 26(1):1–20, 2009.
- [29] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5868–5877, 2017.
- [30] Hao Dang and Peter K Allen. Grasp adjustment on novel objects using tactile experience from similar local geometry. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4007–4012. IEEE, 2013.
- [31] P Datsoris and William Palm. Principles on the development of mechanical hands which can manipulate objects by means of active control. 1985.

- [32] Myron A Diftler, JS Mehling, Muhammad E Abdallah, Nicolaus A Radford, Lyndon B Bridgwater, Adam M Sanders, Roger Scott Askew, D Marty Linn, John D Yamokoski, FA Permenter, et al. Robonaut 2-the first humanoid robot in space. In *2011 IEEE international conference on robotics and automation*, pages 2178–2183. IEEE, 2011.
- [33] Jack Doerner. Fast poisson reconstruction in python. <https://gist.github.com/jackdoerner/b9b5e62a4c3893c76e4c>, 2014.
- [34] Siyuan Dong, Devesh K Jha, Diego Romeres, Sangwoon Kim, Daniel Nikovski, and Alberto Rodriguez. Tactile-rl for insertion: Generalization to objects of unknown geometry. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6437–6443. IEEE, 2021.
- [35] Siyuan Dong, Daolin Ma, Elliott Donlon, and Alberto Rodriguez. Maintaining grasps within slipping bound by monitoring incipient slip. In *IEEE ICRA*, 2018.
- [36] Siyuan Dong, Daolin Ma, Elliott Donlon, and Alberto Rodriguez. Maintaining grasps within slipping bounds by monitoring incipient slip. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3818–3824. IEEE, 2019.
- [37] Siyuan Dong and Alberto Rodriguez. Tactile-based insertion for dense box-packing. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.
- [38] Elliott Donlon, Siyuan Dong, Melody Liu, Jianhua Li, Edward Adelson, and Alberto Rodriguez. Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1927–1934. IEEE, 2018.
- [39] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.
- [40] Bin Fang, Xingming Long, Yifan Zhang, GuoYi Luo, Fuchun Sun, and Huaping Liu. Fabric defect detection using vision-based tactile sensor. *arXiv preprint arXiv:2003.00839*, 2020.
- [41] Nima Fazeli, Miquel Oller, Jiajun Wu, Zheng Wu, Joshua B Tenenbaum, and Alberto Rodriguez. See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion. *Science Robotics*, 4(26):eaav3123, 2019.
- [42] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017.

- [43] Charles Fox, Mat Evans, Martin Pearson, and Tony Prescott. Tactile slam with a biomimetic whiskered robot. In *2012 IEEE International Conference on Robotics and Automation*, pages 4925–4930. IEEE, 2012.
- [44] Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards. *arXiv preprint arXiv:1511.07404*, 2015.
- [45] MI Frecker, GK Ananthasuresh, S Nishiwaki, N Kikuchi, and S Kota. Topological synthesis of compliant mechanisms using multi-criteria optimization. *Journal of Mechanical design*, 119(2):238–245, 1997.
- [46] Aditya Ganapathi, Priya Sundareshan, Brijen Thananjeyan, Ashwin Balakrishna, Daniel Seita, Jennifer Grannen, Minh Hwang, Ryan Hoque, Joseph E. Gonzalez, Nawid Jamali, Katsu Yamane, Soshi Iba, and Ken Goldberg. Learning to smooth and fold real fabric using dense object descriptors trained on synthetic color images, 2020.
- [47] Aditya Ganapathi, Priya Sundareshan, Brijen Thananjeyan, Ashwin Balakrishna, Daniel Seita, Ryan Hoque, Joseph E Gonzalez, and Ken Goldberg. Mmgd: Multi-modal gaussian shape descriptors for correspondence matching in 1d and 2d deformable objects. *arXiv preprint arXiv:2010.04339*, 2020.
- [48] Daniel Fernandes Gomes, Zhonglin Lin, and Shan Luo. Geltip: A finger-shaped optical tactile sensor for robotic manipulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9903–9909. IEEE, 2020.
- [49] Nagamanikandan Govindan and Asokan Thondiyath. Design and analysis of a multimodal grasper having shape conformity and within-hand manipulation with adjustable contact forces. *Journal of Mechanisms and Robotics*, 11(5), 2019.
- [50] Jennifer Grannen, Priya Sundareshan, Brijen Thananjeyan, Jeff Ichnowski, Ashwin Balakrishna, Vainavi Viswanath, Michael Laskey, Joseph Gonzalez, and Ken Goldberg. Learning robot policies for untangling dense knots in linear deformable structures. In *4th Conference on Robot Learning (CoRL)*, 2020.
- [51] Jennifer Grannen, Priya Sundareshan, Brijen Thananjeyan, Jeffrey Ichnowski, Ashwin Balakrishna, Minh Hwang, Vainavi Viswanath, Michael Laskey, Joseph E Gonzalez, and Ken Goldberg. Untangling dense knots by learning task-relevant keypoints. *arXiv preprint arXiv:2011.04999*, 2020.
- [52] Markus Grebenstein, Alin Albu-Schäffer, Thomas Bahls, Maxime Chalon, Oliver Eiberger, Werner Friedl, Robin Gruber, Sami Haddadin, Ulrich Hagn, Robert Haslinger, et al. The dlr hand arm system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3175–3182. IEEE, 2011.

- [53] Randall B Hellman, Cem Tekin, Mihaela van der Schaar, and Veronica J Santos. Functional contour-following via haptic perception and reinforcement learning. *IEEE transactions on haptics*, 11(1):61–72, 2017.
- [54] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [55] Francois R Hogan, Jose Ballester, Siyuan Dong, and Alberto Rodriguez. Tactile dexterity: Manipulation primitives with tactile feedback. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 8863–8869. IEEE, 2020.
- [56] Francois R Hogan, Maria Bauza, Oleguer Canal, Elliott Donlon, and Alberto Rodriguez. Tactile regrasp: Grasp adjustments via simulated tactile transformations. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2963–2970. IEEE, 2018.
- [57] Francois R Hogan, Maria Bauza, Oleguer Canal, Elliott Donlon, and Alberto Rodriguez. Tactile regrasp: Grasp adjustments via simulated tactile transformations. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2963–2970. IEEE, 2018.
- [58] John E Hopcroft, Joseph K Kearney, and Dean B Krafft. A case study of flexible object manipulation. *The International Journal of Robotics Research*, 10(1):41–50, 1991.
- [59] Jonathan B Hopkins and Martin L Culpepper. Synthesis of multi-degree of freedom, parallel flexure system concepts via freedom and constraint topology (fact)–part i: Principles. *Precision Engineering*, 34(2):259–270, 2010.
- [60] Jonathan B Hopkins and Martin L Culpepper. Synthesis of multi-degree of freedom, parallel flexure system concepts via freedom and constraint topology (fact). part ii: Practice. *Precision Engineering*, 34(2):271–278, 2010.
- [61] Ryan Hoque, Daniel Seita, Ashwin Balakrishna, Aditya Ganapathi, Ajay Kumar Tanwani, Nawid Jamali, Katsu Yamane, Soshi Iba, and Ken Goldberg. VisuoSpatial foresight for multi-step, multi-task fabric manipulation, 2020.
- [62] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.
- [63] Larry L Howell. *Compliant mechanisms*. John Wiley & Sons, 2001.
- [64] Larry L Howell, Spencer P Magleby, and Brian M Olsen. *Handbook of compliant mechanisms*. John Wiley & Sons, 2013.
- [65] Larry L Howell and A Midha. A method for the design of compliant mechanisms with small-length flexural pivots. *Journal of mechanical design*, 116(1):280–290, 1994.

- [66] Larry L Howell and Ashok Midha. Parametric deflection approximations for end-loaded, large-deflection beams in compliant mechanisms. *Journal of Mechanical Design*, 117(1):156–165, 1995.
- [67] Gregory Izatt, Geronimo Mirano, Edward Adelson, and Russ Tedrake. Tracking objects with point clouds from vision and touch. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4000–4007. IEEE, 2017.
- [68] Gregory Izatt, Geronimo Mirano, Edward Adelson, and Russ Tedrake. Tracking objects with point clouds from vision and touch. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4000–4007. IEEE, 2017.
- [69] Steve C Jacobsen, John E Wood, DF Knutti, and Klaus B Biggers. The utah/mit dextrous hand: Work in progress. *The International Journal of Robotics Research*, 3(4):21–50, 1984.
- [70] Jasper Wollaston James, Nicholas Pestell, and Nathan F Lepora. Slip detection with a biomimetic tactile sensor. *IEEE Robotics and Automation Letters*, 3(4):3340–3346, 2018.
- [71] Shervin Javdani, Sameep Tandon, Jie Tang, James O’Brien, and Pieter Abbeel. Modeling and perception of deformable one-dimensional objects. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011.
- [72] Jiaqi Jiang, Guanqun Cao, Daniel Fernandes Gomes, and Shan Luo. Vision-guided active tactile perception for crack detection and reconstruction. In *2021 29th Mediterranean Conference on Control and Automation (MED)*, pages 930–936. IEEE, 2021.
- [73] Xin Jiang, Yuki Nagaoka, Kazushi Ishii, Satoko Abiko, Teppei Tsujita, and Masaru Uchiyama. Robotized recognition of a wire harness utilizing tracing operation. *Robotics and Computer-Integrated Manufacturing*, 34:52–61, 2015.
- [74] Atsushi Kakogawa, Hiroyuki Nishimura, and Shugen Ma. Underactuated modular finger with pull-in mechanism for a robotic gripper. In *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 556–561. IEEE, 2016.
- [75] Lydia E. Kavradi, Petr Švestka, Jean Claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 1996.
- [76] Raj Kolamuri, Zilin Si, Yufan Zhang, Arpit Agarwal, and Wenzhen Yuan. Improving grasp stability with rotation measurement from tactile sensing. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6809–6816. IEEE, 2021.

- [77] Naveen Kuppaswamy, Alex Alspach, Avinash Uttamchandani, Sam Creasey, Takuya Ikeda, and Russ Tedrake. Soft-bubble grippers for robust and perceptive manipulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9917–9924. IEEE, 2020.
- [78] Naveen Kuppaswamy, Alex Alspach, Avinash Uttamchandani, Sam Creasey, Takuya Ikeda, and Russ Tedrake. Soft-bubble grippers for robust and perceptive manipulation. *arXiv preprint arXiv:2004.03691*, 2020.
- [79] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020.
- [80] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020.
- [81] Friedrich Lange, Patrick Wunsch, and Gerhard Hirzinger. Predictive vision based control of high speed industrial robot paths. In *1998 IEEE/RSJ International Conference on Robotics and Automation (ICRA)*. IEEE, 1998.
- [82] S M LaValle. Rapidly-Exploring Random Trees: A New Tool for Path Planning. *In*, 1998.
- [83] Susan J Lederman and Roberta L Klatzky. Hand movements: A window into haptic object recognition. *Cognitive psychology*, 19(3):342–368, 1987.
- [84] Michelle A Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. 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)*, pages 8943–8950. IEEE, 2019.
- [85] Nathan F Lepora, Alex Church, Conrad De Kerckhove, Raia Hadsell, and John Lloyd. From pixels to percepts: Highly robust edge perception and contour following using deep learning and an optical biomimetic tactile sensor. *IEEE Robotics and Automation Letters*, 4(2):2101–2107, 2019.
- [86] Nathan F Lepora, Alex Church, Conrad De Kerckhove, Raia Hadsell, and John Lloyd. From pixels to percepts: Highly robust edge perception and contour following using deep learning and an optical biomimetic tactile sensor. *IEEE Robotics and Automation Letters*, 4(2):2101–2107, 2019.

- [87] Adam Lerer, Sam Gross, and Rob Fergus. Learning physical intuition of block towers by example. *arXiv preprint arXiv:1603.01312*, 2016.
- [88] Rui Li, Robert Platt, Wenzhen Yuan, Andreas ten Pas, Nathan Roscup, Mandayam A Srinivasan, and Edward Adelson. Localization and manipulation of small parts using gelsight tactile sensing. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3988–3993. IEEE, 2014.
- [89] Rui Li, Robert Platt, Wenzhen Yuan, Andreas ten Pas, Nathan Roscup, Mandayam A Srinivasan, and Edward Adelson. Localization and manipulation of small parts using gelsight tactile sensing. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3988–3993. IEEE, 2014.
- [90] Huaping Liu, Di Guo, and Fuchun Sun. Object recognition using tactile measurements: Kernel sparse coding methods. *IEEE Transactions on Instrumentation and Measurement*, 65(3):656–665, 2016.
- [91] Wen Hao Lui and Ashutosh Saxena. Tangled: Learning to untangle ropes with rgb-d perception. In *Intelligent Robots and Systems (IROS), 2013 IEEE International Conference on*, pages 837–844. IEEE, 2013.
- [92] Shan Luo, Joao Bimbo, Ravinder Dahiya, and Hongbin Liu. Robotic tactile perception of object properties: A review. *Mechatronics*, 48:54–67, 2017.
- [93] Shan Luo, Xiaozhou Liu, Kaspar Althoefer, and Hongbin Liu. Tactile object recognition with semi-supervised learning. In *International Conference on Intelligent Robotics and Applications*, pages 15–26. Springer, 2015.
- [94] Shan Luo, Wenzhen Yuan, Edward Adelson, Anthony G Cohn, and Raul Fuentes. Vitac: Feature sharing between vision and tactile sensing for cloth texture recognition. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2722–2727. IEEE, 2018.
- [95] Daolin Ma, Elliott Donlon, Siyuan Dong, and Alberto Rodriguez. Dense tactile force estimation using gelslim and inverse fem. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5418–5424. IEEE, 2019.
- [96] Raymond R Ma and Aaron M Dollar. An underactuated hand for efficient finger-gaiting-based dexterous manipulation. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pages 2214–2219. IEEE, 2014.
- [97] Raymond R Ma and Aaron M Dollar. In-hand manipulation primitives for a minimal, underactuated gripper with active surfaces. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 50152, page V05AT07A072. American Society of Mechanical Engineers, 2016.

- [98] Raymond R Ma, Nicolas Rojas, and Aaron M Dollar. Spherical hands: Toward underactuated, in-hand manipulation invariant to object size and graspr location. *Journal of Mechanisms and Robotics*, 8(6):061021, 2016.
- [99] Uriel Martinez-Hernandez, Giorgio Metta, Tony J Dodd, Tony J Prescott, Lorenzo Natale, and Nathan F Lepora. Active contour following to explore object shape with robot touch. In *2013 World Haptics Conference (WHC)*, pages 341–346. IEEE, 2013.
- [100] Matthew T Mason and Kevin M Lynch. Dynamic manipulation. In *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, volume 1, pages 152–159. IEEE, 1993.
- [101] Hermann Mayer, Faustino Gomez, Daan Wierstra, Istvan Nagy, Alois Knoll, and Jürgen Schmidhuber. A system for robotic heart surgery that learns to tie knots using recurrent neural networks. *Advanced Robotics*, 22(13-14):1521–1537, 2008.
- [102] Connor M McCann and Aaron M Dollar. Design of a stewart platform-inspired dexterous hand for 6-dof within-hand manipulation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1158–1163. IEEE, 2017.
- [103] Dale McConachie, Andrew Dobson, Mengyao Ruan, and Dmitry Berenson. Manipulating deformable objects by interleaving prediction, planning, and control. *International Journal of Robotics Research*, 2020.
- [104] Mark Moll and Lydia Kavraki. Path planning for deformable linear objects. *IEEE Transactions on Robotics*, pages 625–636, 2006.
- [105] Takuma Morita, Jun Takamatsu, Koichi Ogawara, Hiroshi Kimura, and Katsushi Ikeuchi. Knot planning from observation. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 3, pages 3887–3892. IEEE, 2003.
- [106] Tyler Morrison and Hai-Jun Su. Stiffness modeling of a variable stiffness compliant link. *Mechanism and Machine Theory*, 153:104021, 2020.
- [107] Ashvin Nair, Dian Chen, Pulkit Agrawal, Phillip Isola, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Combining self-supervised learning and imitation for vision-based rope manipulation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2146–2153. IEEE, 2017.
- [108] Allison M Okamura and Mark R Cutkosky. Feature detection for haptic exploration with robotic fingers. *The International Journal of Robotics Research*, 20(12):925–938, 2001.

- [109] Akhil Padmanabha, Frederik Ebert, Stephen Tian, Roberto Calandra, Chelsea Finn, and Sergey Levine. Omnitact: A multi-directional high resolution touch sensor. *arXiv preprint arXiv:2003.06965*, 2020.
- [110] Antoine Petit, Vincenzo Lippiello, and Bruno Siciliano. Real-time tracking of 3d elastic objects with an rgb-d sensor. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3914–3921. IEEE, 2015.
- [111] C Piazza, G Grioli, MG Catalano, and AJAROC Bicchi. A century of robotic hands. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:1–32, 2019.
- [112] Branden Romero, Filipe Veiga, and Edward Adelson. Soft, round, high resolution tactile fingertip sensors for dexterous robotic manipulation. *arXiv preprint arXiv:2005.09068*, 2020.
- [113] Mitul Saha, Pekka Isto, and Jean-Claude Latombe. Motion planning for robotic manipulation of deformable linear objects. *Experimental Robotics*, page 23–32, 2008.
- [114] Khairul Salleh Mohamed Sahari, Hiroaki Seki, Yoshitsugu Kamiya, and Masatoshi Hikizu. Edge tracing manipulation of clothes based on different gripper types. *Journal of Computer Science*, 2010.
- [115] JK Salisbury. Kinematics and force analysis of articulated hands, phd thesis, 1982.
- [116] Alexander Schmitz, Yusuke Bansho, Kuniaki Noda, Hiroyasu Iwata, Tetsuya Ogata, and Shigeki Sugano. Tactile object recognition using deep learning and dropout. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 1044–1050. IEEE, 2014.
- [117] Carmelo Sferrazza, Thomas Bi, and Raffaello D’Andrea. Learning the sense of touch in simulation: a sim-to-real strategy for vision-based tactile sensing. *arXiv preprint arXiv:2003.02640*, 2020.
- [118] Carmelo Sferrazza and Raffaello D’Andrea. Design, motivation and evaluation of a full-resolution optical tactile sensor. *Sensors*, 19(4):928, 2019.
- [119] Yu She, Shaoxiong Wang, Siyuan Dong, Neha Sunil, Alberto Rodriguez, and Edward Adelson. Cable manipulation with a tactile-reactive gripper. *arXiv preprint arXiv:1910.02860*, 2019.
- [120] Yu She, Shaoxiong Wang, Siyuan Dong, Neha Sunil, Alberto Rodriguez, and Edward Adelson. Cable manipulation with a tactile-reactive gripper. *The International Journal of Robotics Research*, 40(12-14):1385–1401, 2021.

- [121] Jian Shi, J Zachary Woodruff, Paul B Umbanhowar, and Kevin M Lynch. Dynamic in-hand sliding manipulation. *IEEE Transactions on Robotics*, 33(4):778–795, 2017.
- [122] Kazuhiro Shimonomura. Tactile image sensors employing camera: A review. *Sensors*, 19(18):3933, 2019.
- [123] Avishai Sintov, Or Tslil, and Amir Shapiro. Robotic swing-up regrasping manipulation based on the impulse–momentum approach and clqr control. *IEEE Transactions on Robotics*, 32(5):1079–1090, 2016.
- [124] Edward Smith, Roberto Calandra, Adriana Romero, Georgia Gkioxari, David Meger, Jitendra Malik, and Michal Drozdal. 3d shape reconstruction from vision and touch. *Advances in Neural Information Processing Systems*, 33:14193–14206, 2020.
- [125] Nicolas Sommer, Miao Li, and Aude Billard. Bimanual compliant tactile exploration for grasping unknown objects. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6400–6407. IEEE, 2014.
- [126] Simon Stepputtis, Yezhou Yang, and Heni Ben Amor. Extrinsic dexterity through active slip control using deep predictive models. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3180–3185. IEEE, 2018.
- [127] Priya Sundareshan, Jennifer Grannen, Brijen Thananjeyan, Ashwin Balakrishna, Michael Laskey, Kevin Stone, Joseph E Gonzalez, and Ken Goldberg. Learning rope manipulation policies using dense object descriptors trained on synthetic depth data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9411–9418. IEEE, 2020.
- [128] Sudharshan Suresh, Maria Bauza, Kuan-Ting Yu, Joshua G Mangelson, Alberto Rodriguez, and Michael Kaess. Tactile slam: Real-time inference of shape and pose from planar pushing. *arXiv preprint arXiv:2011.07044*, 2020.
- [129] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [130] Te Tang, Yongxiang Fan, Hsien-Chung Lin, and Masayoshi Tomizuka. State estimation for deformable objects by point registration and dynamic simulation. In *Intelligent Robots and Systems (IROS), 2017 IEEE International Conference on*. IEEE, 2017.
- [131] Stephen Tian, Frederik Ebert, Dinesh Jayaraman, Mayur Mudigonda, Chelsea Finn, Roberto Calandra, and Sergey Levine. Manipulation by feel: Touch-based control with deep predictive models. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 818–824. IEEE, 2019.

- [132] Stephen Tian, Frederik Ebert, Dinesh Jayaraman, Mayur Mudigonda, Chelsea Finn, Roberto Calandra, and Sergey Levine. Manipulation by feel: Touch-based control with deep predictive models. *arXiv preprint arXiv:1903.04128*, 2019.
- [133] Vinicio Tincani, Manuel G Catalano, Edoardo Farnioli, Manolo Garabini, Giorgio Grioli, Gualtiero Fantoni, and Antonio Bicchi. Velvet fingers: A dexterous gripper with active surfaces. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1257–1263. IEEE, 2012.
- [134] N Tolou and JL Herder. A semianalytical approach to large deflections in compliant beams under point load. *Mathematical Problems in Engineering*, 2009, 2009.
- [135] Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 2442–2447. IEEE, 2017.
- [136] Filipe Veiga, Jan Peters, and Tucker Hermans. Grip stabilization of novel objects using slip prediction. *IEEE transactions on haptics*, 11(4):531–542, 2018.
- [137] Filipe Veiga, Herke Van Hoof, Jan Peters, and Tucker Hermans. Stabilizing novel objects by learning to predict tactile slip. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5065–5072. IEEE, 2015.
- [138] Chen Wang, Shaoxiong Wang, Branden Romero, Filipe Veiga, and Edward Adelson. Swingbot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020.
- [139] Fei Wang, Etienne Burdet, Ronald Vuillemin, and Hannes Bleuler. Knot-tying with visual and force feedback for vr laparoscopic training. In *Engineering in Medicine and Biology 27th Annual Conference*. IEEE, 2005.
- [140] Shaoxiong Wang, Yu She, Branden Romero, and Edward Adelson. Gelsight wedge: Measuring high-resolution 3d contact geometry with a compact robot finger. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6468–6475. IEEE, 2021.
- [141] Shaoxiong Wang, Jiajun Wu, Xingyuan Sun, Wenzhen Yuan, William T Freeman, Joshua B Tenenbaum, and Edward H Adelson. 3d shape perception from monocular vision, touch, and shape priors. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1606–1613. IEEE, 2018.

- [142] Weifu Wang, Dmitry Berenson, and Devin Balkcom. An online method for tight-tolerance insertion tasks for string and rope. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2488–2495. IEEE, 2015.
- [143] Benjamin Ward-Cherrier, Nicholas Pestell, Luke Cramphorn, Benjamin Winstone, Maria Elena Giannaccini, Jonathan Rossiter, and Nathan F Lepora. The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies. *Soft robotics*, 5(2):216–227, 2018.
- [144] Benjamin Ward-Cherrier, Nicholas Pestell, Luke Cramphorn, Benjamin Winstone, Maria Elena Giannaccini, Jonathan Rossiter, and Nathan F Lepora. The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies. *Soft robotics*, 5(2):216–227, 2018.
- [145] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.
- [146] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. *Advances in neural information processing systems*, 30, 2017.
- [147] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Advances in neural information processing systems*, pages 127–135, 2015.
- [148] Yilin Wu, Wilson Yan, Thanard Kurutach, Lerrel Pinto, and Pieter Abbeel. Learning to manipulate deformable objects without demonstrations, 2019.
- [149] Zhenjia Xu, Jiajun Wu, Andy Zeng, Joshua B Tenenbaum, and Shuran Song. Densephysnet: Learning dense physical object representations via multi-step dynamic interactions. *arXiv preprint arXiv:1906.03853*, 2019.
- [150] 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)*, pages 1045–1051. IEEE, 2016.
- [151] 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)*, pages 1045–1051. IEEE, 2016.
- [152] Yuji Yamakawa, Akio Namiki, and Masatoshi Ishikawa. Motion planning for dynamic knotting of a flexible rope with a high-speed robot arm. In *Intelligent Robots and Systems (IROS), 2010 IEEE International Conference on*, pages 49–54. IEEE, 2010.

- [153] Yuji Yamakawa, Akio Namiki, and Masatoshi Ishikawa. Simple model and deformation control of a flexible rope using constant, high-speed motion of a robot arm. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2012.
- [154] Yuji Yamakawa, Akio Namiki, Masatoshi Ishikawa, and Makoto Shimojo. One-handed knotting of a flexible rope with a high-speed multifingered hand having tactile sensors. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2007.
- [155] Mengyuan Yan, Yilin Zhu, Ning Jin, and Jeannette Bohg. Self-supervised learning of state estimation for manipulating deformable linear objects. *arXiv preprint arXiv:1911.06283*, 2019.
- [156] Lin Yen-Chen, Maria Bauza, and Phillip Isola. Experience-embedded visual foresight. *arXiv preprint arXiv:1911.05071*, 2019.
- [157] Zhengkun Yi, Roberto Calandra, Filipe Veiga, Herke van Hoof, Tucker Hermans, Yilei Zhang, and Jan Peters. Active tactile object exploration with gaussian processes. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4925–4930. IEEE, 2016.
- [158] Hanna Yousef, Mehdi Boukallel, and Kaspar Althoefer. Tactile sensing for dexterous in-hand manipulation in robotics—a review. *Sensors and Actuators A: physical*, 167(2):171–187, 2011.
- [159] Kuan-Ting Yu, Maria Bauza, Nima Fazeli, and Alberto Rodriguez. More than a million ways to be pushed. a high-fidelity experimental dataset of planar pushing. In *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 30–37. IEEE, 2016.
- [160] Shenli Yuan, Austin D. Epps, Jerome B. Nowak, and J. Kenneth Salisbury. Design of a roller-based dexterous hand for object grasping and within-hand manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8870–8876, 2020.
- [161] Shenli Yuan, Lin Shao, Connor L. Yako, Alex Gruebele, and J. Kenneth Salisbury. Design and control of roller grasper v2 for in-hand manipulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9151–9158, 2020.
- [162] Wenzhen Yuan, Siyuan Dong, and Edward Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.
- [163] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.

- [164] Wenzhen Yuan, Rui Li, Mandayam A Srinivasan, and Edward H Adelson. Measurement of shear and slip with a gelsight tactile sensor. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 304–311. IEEE, 2015.
- [165] Wenzhen Yuan, Yuchen Mo, Shaoxiong Wang, and Edward H Adelson. Active clothing material perception using tactile sensing and deep learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4842–4849. IEEE, 2018.
- [166] Wenzhen Yuan, Shaoxiong Wang, Siyuan Dong, and Edward Adelson. Connecting look and feel: Associating the visual and tactile properties of physical materials. *arXiv preprint arXiv:1704.03822*, 2017.
- [167] Hiroyuki Yuba, Solvi Arnold, and Kimitoshi Yamazaki. Unfolding of a rectangular cloth from unarranged starting shapes by a Dual-Armed robot with a mechanism for managing recognition error and uncertainty. *Advanced Robotics*, 2017.
- [168] Shigang Yue and Dominik Henrich. Manipulating deformable linear objects: Sensor-based fast manipulation during vibration. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2002.
- [169] Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *arXiv preprint arXiv:1903.11239*, 2019.
- [170] Guanlan Zhang, Yipai Du, Hongyu Yu, and Michael Yu Wang. Deltact: A vision-based tactile sensor using dense color pattern. *arXiv preprint arXiv:2202.02179*, 2022.
- [171] Harry Zhang, Jeffrey Ichnowski, Daniel Seita, Jonathan Wang, and Ken Goldberg. Robots of the lost arc: Learning to dynamically manipulate fixed-endpoint ropes and cables. *arXiv preprint arXiv:2011.04840*, 2010.
- [172] Jihong Zhu, Benjamin Navarro, Philippe Fraitse, André Crosnier, and Andrea Cherubini. Dual-arm robotic manipulation of flexible cables. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 479–484. IEEE, 2018.
- [173] Jihong Zhu, Benjamin Navarro, Robin Passama, Philippe Fraitse, André Crosnier, and Andrea Cherubini. Robotic manipulation planning for shaping deformable linear objects with environmental contacts. *IEEE Robotics and Automation Letters*, 5(1):16–23, 2019.
- [174] Liang Zou, Chang Ge, Z Jane Wang, Edmond Cretu, and Xiaoou Li. Novel tactile sensor technology and smart tactile sensing systems: A review. *Sensors*, 17(11):2653, 2017.