TEXterity: Tactile Extrinsic deXterity

by

Sangwoon Kim

B.S. Mechanical and Aerospace Engineering, Seoul National University, 2018
M.S. Mechanical Engineering, MIT, 2020

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024

Authored by:	Sangwoon Kim Department of Mechanical Engineering May xx, 2024
Certified by:	Alberto Rodriguez Visiting Scientist, Thesis Supervisor
Accepted by:	Nicolas Hadjiconstantinou Professor of Mechanical Engineering Graduate Officer, Department of Mechanical Engineering

TEXterity: Tactile Extrinsic deXterity

by

Sangwoon Kim

Submitted to the Department of Mechanical Engineering
on May xx, 2024 in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN MECHANICAL ENGINEERING

ABSTRACT

This thesis introduces the concept of *TEXterity* (***T***actile ***E***xtrinsic de***X***terity) to address challenges in robotic manipulation. Focusing on tactile sensing, TEXterity aims to enhance dexterity by enabling robots to perceive and act upon extrinsic contact between the grasped object and the environment.

Identifying interpretability, observability, and uncertainty as key challenges in tactile sensing, this thesis sets out to answer four pivotal questions:

- *Is tactile sensing actually useful?*: Chapter 2 explores the practical utility of tactile sensing through an examination of its application in a peg-in-hole insertion task. It demonstrates the advantages of tactile sensing over conventional force-torque sensing.
- *How can we exploit tactile sensing efficiently?*: Chapter 3 proposes an efficient approach to exploit tactile sensing by introducing extrinsic contact as an interpretable representation. This method offers scalability and effectiveness in utilizing tactile data.
- *How can we reason about extrinsic contact with tactile sensing?*: Chapter 4 develops a novel method for simultaneous estimation and control of extrinsic contact states, addressing uncertainties introduced by physical interactions. It enables robots to reason effectively about extrinsic contact using tactile sensing in a controlled manner.
- *How can we achieve extrinsic dexterity with tactile sensing?*: Chapter 5 extends the simultaneous estimation and control method to achieve extrinsic dexterity, showcasing precise in-hand sliding regrasps facilitated by pushing the object against the external environment.

The conclusion summarizes the key findings, emphasizing the significance of tactile sensing and TEXterity in addressing challenges and advancing robotic manipulation. Strategies to tackle major challenges are outlined, focusing on interpretability, observability, and uncertainty.

In essence, this thesis lays the groundwork for unlocking the potential of tactile sensing in robotic manipulation, offering insights, solutions, and avenues for future research to propel the field toward achieving TEXterity and further toward human-level dexterity.

Please see real-world demonstration videos at this url.

Thesis supervisor: Alberto Rodriguez
Title: Visiting Scientist

Acknowledgments

I am extremely grateful to my advisor, *Alberto Rodriguez*, for his mentorship and support throughout my Ph.D. He has been the person I can turn to for advice not only regarding research directions but also when navigating various slumps and challenges throughout my Ph.D. While I empathize with the MechE department for losing such an outstanding faculty member, I look forward to witnessing the great achievements he will make in his new adventure at Boston Dynamics.

I also thank dear committee members, *John Leonard*, *Pulkit Agrawal*, and *Kim Sangbae*, for taking their time and effort to review my work and provide constructive feedback. Their invaluable insights have significantly broadened my perspective, allowing me to scrutinize aspects of my research that may have eluded my attention. The feedback has been instrumental in fostering a more comprehensive and objective understanding of my work, and I appreciate their contributions to the refinement of my research endeavors.

All of my achievements would not have been possible without help from my labmates and our administrative assistant, *Lisa Maxwell.* Although we faced the challenge of COVID-19 at the beginning of joining the MCube lab, I received so much support from the lab. I am thankful to *Siyuan Dong* for being the closest mentor in my first year in the lab. The work that I had done with him in the first year has shaped my research direction throughout my PhD journey. I thank *Neel Doshi*, for the support that he provided both at MCube lab and also during my internship at Amazon Robotics. I appreciate *Ian Taylor*'s help in setting up all the research equipment and providing relief when *Antonia Bronars* and I were panicking from broken sensors. I am thankful to *Antonia Bronars* for being the closest collaborator during my final project at MIT. I wish her all the best for the remaining years of her Ph.D.

I thank my collaborators and sponsors, especially *Devesh Jha* and *Diego Romeres* from *MERL*, and *Parag Patre* from *Magna International.* The discussion that we had has honed my research to the level of industry standard and made me think carefully about how my research can be utilized to create real value in the future.

I am also grateful to the robots of the MCube lab, particularly the old ABB IRB 120. While it does not have any emotion, I spent so much time with it that I feel emotionally attached to it. I heard the ABB will collect it back and maybe recycle it after the impending closedown of the MCube lab. Wherever it goes after the recycling, I hope it brings the same level of joy and value that I have gained through using this robot to whoever will use this robot.

My Ph.D. journey would not have been the same without all the friends and peers that I got to know during my grad study at MIT, Boston, and conferences. I especially thank the people of *KGSA (Korean Graduate Student Association)* and *KGSA-MechE.* Life as an alien in the US is challenging, especially when the home is 14 hours away by flight. People of *KGSA* made Boston feel like home, alleviating my homesickness toward Seoul and providing support through various activities, such as the qualification exam rehearsal.

I am extremely thankful to my direct family – *Kim Uoong Chul*, *Hong Eunsook*, *Kim Jia*, and *Kim Jiyun* – and extended family for their endless support. Leaving my parents and standing on my own feet in a foreign country was a challenge not only for me but also for my parents. Nevertheless, they have been determined to overcome this challenge and sent me enormous support across the Pacific Ocean. I hope they stay healthy and happy for a very very long time.

Last but not least, I am eternally grateful to my fiancée, *Kim Hyunkyung*. She is not only my partner and soulmate but also my go-to person whenever I face various life challenges, whether it's deciding what to have for dinner or making crucial career decisions like choosing which job offer to accept. Without such emotional support, I doubt whether I could have come this far in my journey. I appreciate the moments we have shared so far and eagerly anticipate all that is yet to come.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation:

What is *TEXterity* and Why We Need It?

With only several years since the monumental milestone of AlphaGo [1] in the history of artificial intelligence (AI), AI has rapidly progressed, manifesting breakthroughs that extend beyond virtual realms to impact the physical world. For example, legged robots trained primarily in simulation can ascend a real mountain [2] and perform parkour [3], and AI-driven pilots outperform human champions in drone racing [4]. These examples exemplify the tangible strides made in the field of autonomy and robotics. That said, within the broader scope of robotics, the domain of robotic manipulation appears to lag behind compared to advancements in locomotion and perception, particularly in tasks that require reasoning and controlling physical contact interactions.

Until very recently, mainstream research in robotic manipulation focused on pick-and-place types of tasks, which require a relatively less meticulous understanding of physical contact interactions. For example, the recent emergence of large language model (LLM) or large multimodal model (LMM)-based methods in robotic manipulation has shown promise in achieving unprecedented levels of generalization [5]–[7]. However, these methods still have inherent limitations in performing more intricate physical interactions – such as screwing a bolt, spreading jam on bread, or peeling potatoes – since their models primarily rely only on vision and language information, lacking information on embodied physical interactions like force-torque sensing or tactile sensing.

Addressing the challenges of robotic manipulation to go beyond the pick-and-place tasks necessitates a deeper understanding and integration of physical contact interactions. Several approaches have been pursued, such as modeling complex physical interactions in simulation and training robot behavior within the simulated environment [8], [9]. Another approach involves employing model-based optimization for motion planning [10]. Recently gaining attention is the approach of teaching robots behaviors through human demonstrations [11]–[13]. While these approaches show promise, they have yet to achieve human-level dexterity.

This thesis contends that, regardless of the chosen approach, a fundamental requirement for advancements in robotic manipulation is a comprehensive understanding and strategic utilization of physical contact interactions. The need to bridge the gap between current capabilities and the complexity of real-world physical interaction is evident. The central question emerges: How can we empower robots to better sense, understand, and act upon physical contact?

To make robots embrace physical contact interaction, we focus on tactile sensing. Humans rely on tactile feedback in daily activities like playing guitar, writing with a pencil, and lighting a match, just to name a few. This highlights the importance of imparting this ability to leverage tactile sensing to robots for human-level dexterity. In this thesis, we assert that embracing physical contact involves not only sensing but also comprehending and acting in response to tactile stimuli. In this context, we introduce the concept of *TEXterity*, denoting ***T***actile ***E***xtrinsic de***X***terity, as a novel direction to propel advancements in robotic manipulation.

When manipulating an object with fingers, contact interaction can be primarily classified into two categories, as depicted in Figure 1.1. First, there is intrinsic contact between the

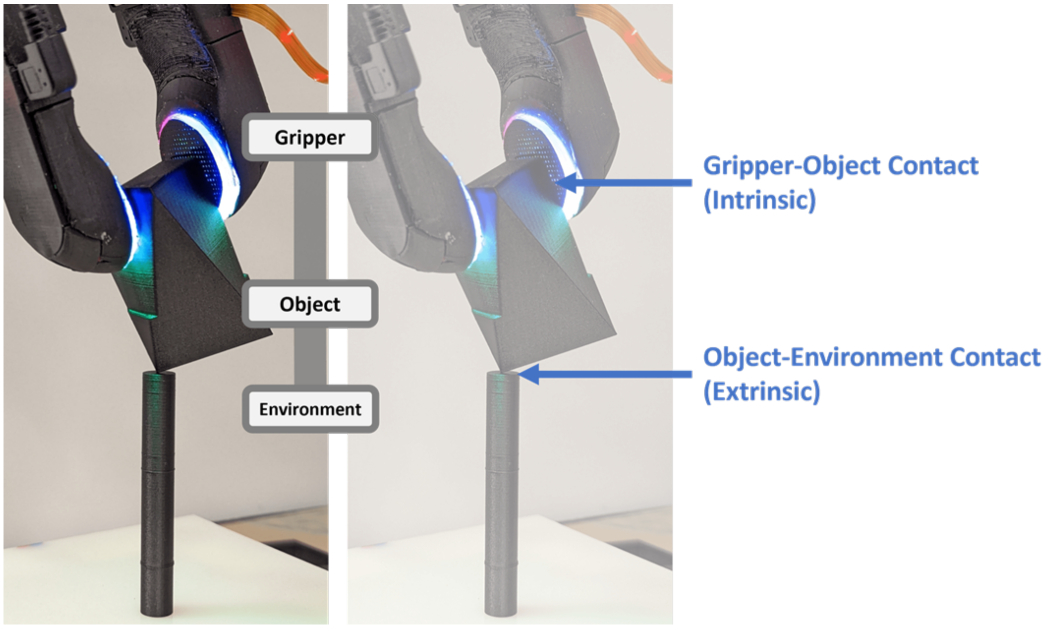


Figure 1.1: Intrinsic and Extrinsic Contact

finger and the grasped object. There is also extrinsic contact between the grasped object and an extrinsic environment. While it is integral in many object manipulation tasks to understand and control the extrinsic contact interaction, it is challenging since extrinsic contacts cannot be directly sensed. Instead, it has to be indirectly inferred through the chain of contacts that connects the finger to the object and to the extrinsic environment. TEXterity aims to harness tactile sensing to interact with the extrinsic environment through extrinsic contact, subsequently leveraging this interaction to enhance dexterous manipulation.

1.2 Challenges

Being a relatively new sensing modality compared to vision or other various encoders, there are major challenges to be solved in order to fully leverage tactile sensing and further achieve TEXterity:

- ***Interpretability***: Unlike other conventional sensing modalities, there is no unified protocol for interpreting tactile sensing. There are many different form factors, and even within the similar form factor, there is no standard practice on how to efficiently process and utilize tactile information. For example, vision-based tactile sensors like GelSight [14], observe deformation on its finger as a high-resolution image. However, it remains unclear how to efficiently interpret and leverage this high-dimensional information as input to the robots.

- ***Observability***: Tactile sensors provide local information on where they are touching, but not elsewhere. This poses challenges in reasoning the state of the system globally. Specifically, this makes it challenging to reason about the contact interaction occurring at the location where the finger is not directly touching, i.e., extrinsic contact.

- ***Uncertainty***: Tactile sensing is inherently linked with physical contact, exerting a tangible impact on the system and introducing added uncertainty. When a tactile sensor makes contact with an object, it not only provides information about the system but also, concurrently, influences the system due to the physical interaction with the finger, introducing a unique challenge of uncertainty. This inherent challenge sets tactile sensing apart from vision sensing using an external camera, which merely observes the system without exerting any physical influence. The effective resolution of this augmented uncertainty becomes pivotal for ensuring the reliable utilization of tactile feedback.

1.3 Contributions

This thesis compiles previous publications[1] aimed at addressing challenges in tactile sensing and achieving TEXterity. We tackle these challenges by exploring consecutive questions:

- Is tactile sensing actually useful? (Chapter 2 [18])

- How can we exploit tactile sensing more efficiently? (Chapter 3 [15])

[1] The author of this thesis is listed as the first or co-first author in [15]–[17], and as a co-author in [18]

- How can we reason about extrinsic contact interaction with tactile sensing? (Chapter 4 [16])
- How can we achieve extrinsic dexterity with tactile sensing (Chapter 5 [17])

In Chapter 2 [18], we justify our use of tactile sensing over other sensing modalities like force-torque (F/T) sensing by investigating its utility. We train an end-to-end reinforcement learning (RL) policy for a peg-in-hole insertion task, where the robot takes input as raw tactile sensing and outputs a motion to correct the misalignment between the peg and the hole. Our results demonstrate that the policy trained with tactile sensing outperforms the one trained with F/T sensing. Additionally, we provide a detailed discussion on the advantages of using tactile sensing.

While we show in Chapter 2 [18] that tactile sensing is useful, we discover that the training process is tedious, requiring 8 hours of real robot experiments, hindering the scalability of the approach. In essence, the end-to-end approach utilizing tactile sensing proves inefficient for scalability. Therefore, in Chapter 3 [15], we delve into a more efficient strategy for leveraging tactile sensing. Adopting interpretable abstractions, such as extrinsic contact location, as an intermediate representation, allows us to train the peg-in-hole insertion policy without any real experiments and still outperform the end-to-end approach. In other words, our exploration highlights that extrinsic contact sensing provides an efficient means of representing contact interaction, paving the way for a more efficient, interpretable, and scalable method for utilizing tactile sensing.

However, the preliminary extrinsic contact sensing introduced in Chapter 3 [15] lacks generalizability across various contact configurations. Thus, in Chapter 4 [16], we devise a method to comprehend the extrinsic contact state across diverse configurations. A critical challenge in this endeavor involves addressing uncertainty introduced by physical interactions due to contacts. While exploratory robot motion provides information to reason about the contact state, it concurrently affects and alters the contact state, adding uncertainty; for instance, the extrinsic contact might slip on a slippery surface during the exploratory gauging motion. To overcome this hurdle, we develop a method capable of simultaneously estimating and controlling the contact state. By employing this simultaneous estimation and control method, we achieve sub-millimeter accuracy in estimating the contact location while effectively maintaining control over the contact state.

While Chapter 4 [16] serves as a framework for estimating and controlling the extrinsic contact state, it lacks an understanding of the grasped object's shape and pose. Consequently, it possesses limited capability to reason about the global reconfiguration of the object, i.e., in-hand manipulation - an essential aspect in tasks like assembly or tool use. In Chapter 5 [17], we address this limitation by integrating the *Tac2Pose* [19] estimator, capable of estimating the relative gripper/pose grasp pose for known but arbitrary objects. This integration enables the development of a method that not only estimates the object's pose and associated contact configurations but also simultaneously controls them. With these capabilities, we successfully address the problem of precisely controlling in-hand sliding regrasps by pushing against an external environment, achieving extrinsic dexterity.

Chapter 2

Tactile-RL for Insertion:

Is Tactile Sensing Actually Useful?

Object insertion is a classic contact-rich manipulation task. The task remains challenging, especially when considering general objects of unknown geometry, which significantly limits the ability to understand the contact configuration between the object and the environment. We study the problem of aligning the object and environment with a tactile-based feedback insertion policy. The insertion process is modeled as an episodic policy that iterates between insertion attempts followed by pose corrections. We explore different mechanisms to learn such a policy based on Reinforcement Learning. The key contribution of this chapter is to demonstrate that it is possible to learn a tactile insertion policy that generalizes across different object geometries, and an ablation study of the key design choices for the learning agent: 1) the type of learning scheme: supervised vs. reinforcement learning; 2) the type of learning schedule: unguided vs. curriculum learning; 3) the type of sensing modality: force/torque vs. tactile; and 4) the type of tactile representation: tactile RGB vs. tactile flow. We show that the optimal configuration of the learning agent (RL + curriculum + tactile flow) exposed to 4 training objects yields a closed-loop insertion policy that inserts 4 novel objects with over 85.0% success rate and within $3 \sim 4$ consecutive attempts. Comparisons between F/T and tactile sensing show that while an F/T-based policy learns more efficiently,

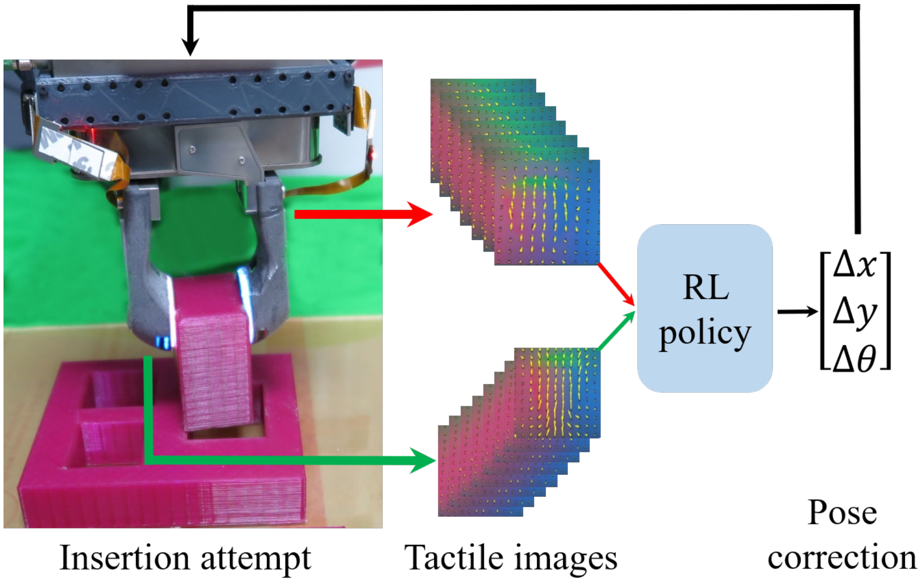

RL policy
$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta\theta\end{bmatrix}$
Pose correction
Insertion attempt
Tactile images

Figure 2.1: Insertion task with the proposed Tactile RL policy.

a tactile-based policy provides better generalization. See supplementary video and results at https://sites.google.com/view/tactileinsertion.

2.1 Introduction

Localizing contacts between objects and their environment is central to many contact-rich robotic manipulation tasks [20]. Consider for example the classical tasks of inserting a peg in a hole or packing an object in a box. A mismatch between the real and modeled object's shape or between the real and target object's pose generates unexpected contacts between the object and the hole, potentially leading to complex contact interlocking configurations [21].

The location of contacts during insertion depends on the geometrical attributes of the object and hole. Techniques to explicitly recover the location of those contacts, usually require knowledge of object geometries and/or are only applicable to limited types of contact configurations [21]–[24]. The estimation of those contacts without knowledge of object geometry is also possible in some cases by aggregating tactile information over time [25].

In this chapter, alternatively, we are interested in feedback-based mechanisms to interactively correct the miss-alignment between object and hole without making use of the geometry of the shapes that generated it and without making explicit inferences of the contact location. Tactile sensors co-located at the gripper fingers (see Figure 2.1) are better positioned to capture contact events than vision sensors, which suffer from occlusions and limited accuracy. Force/Torque (F/T) sensors can also be co-located at fingers or wrists. In this work, we investigate the use of F/T and high-resolution tactile sensors to guide peg-in-hole insertion policies and to generalize to different objects while accounting for variations in contact formations, in grasps, and in environments, i.e., a general *tactile insertion agent*.

The key challenge to a general insertion policy is that, in the absence of geometric models, the contact state is not fully observable from a single time instant, neither for force nor tactile sensing. The insertion agent, then, needs to consider policies beyond greedily trying to estimate and correct the miss-alignment between part and hole, as was shown in previous schemes based on self-supervised learning, which showed limited generalization to objects and contact configurations [21], [26]. In recent work [26], we demonstrated that high-resolution tactile sensors such as GelSlim [27] allow feedback correction for novel objects. However, the simplified insertion environment (a slot with constrain in one axis) only covers a very small set of possible contact formations in an object insertion problem.

In this chapter, we explore the more complex and general problem of *inserting multiple objects into multiple environments with a single policy*, and we ablate several important design choices for an RL tactile-based insertion agent. In particular, we evaluate:

- **Supervised vs. Reinforced** We study the importance of the sequential nature of the learning process in rewarding successful insertions, and learn an episodic RL agent that iterates between insertion attempts and estimating alignment corrections (see Figure 2.1). While the supervised agent suffers in the more constrained insertion environments, the RL agent shows significantly better performance. The RL formulation, with the ability to consider delayed rewards, opens the possibility of searching through non-greedy policies.

- **Curriculum training.** We show that modest guidance of the learned policy to progressively tackle more complex environments (wall → corner → U → hole) is effective at increasing the data efficiency.

- **Tactile representation.** We show that learning an insertion agent from tactile flow (represented with a marker array) yields much better generalization to variations in object and grasp geometry than directly learning from raw tactile RGB images. Tactile flow is effective at preventing overfitting to object surface texture.

Since the dynamics of insertion and the dynamics of tactile sensing are both guided by complex and difficult-to-simulate contact mechanics, the use of real experiments and the need for data efficiency are key drivers of this work. We also compare our tactile-RL agent with a policy trained using F/T sensors and show that the tactile-RL agent provides better generalization to objects of different geometry. These experiments also demonstrate the limitations of F/T measurements that can be alleviated using tactile sensors. All the experiments in this chapter are done in a real system, with real data, and in real-time. The insertion RL agent is learned in 500 episodes (8 hours of robot time). For this purpose, we developed an experimental setup (Figure 2.3) that allows running tactile-based automated insertion experiments under controlled variations on the initial grasp on different objects, and with an automatic resetting mechanism based on object-specific alignment features.

2.2 Related Work

Peg insertion tasks have been studied for a long time, due to their importance in manufacturing. Early work on peg-in-hole insertion algorithms was based on developing passive compliance devices, [28], [29]. These devices, as well as similarly operating linear impedance controllers, can successfully complete insertions with chamfered holes, as long as the initial contact point is on the chamfer. Other methods use model-based approaches to directly estimate the pose of the peg relative to the hole with force feedback [30], [31]. However, these require knowledge of object models and/or are constrained to a single object type. Furthermore, there is a fundamental ambiguity in the mapping from alignment errors to sensed forces — in many contact situations, it is many-to-one, so it cannot be inverted directly [32].

An early RL approach to the actual insertion phase was proposed in [33], similarly based on an episodic RL setting from observations of position and forces from an F/T sensor. Since then, several RL approaches have been proposed, mainly based on sensing and acting on the positional state and force-torques of the robotic manipulators. In [34], time-varying linear-Gaussian controllers combined together via a guided policy search component are presented. [35] studies the insertion of a rigid peg into a deformable hole, where the position of the hole relative to the robot is known, possibly with minor uncertainty. This chapter combines robot-state information and filtered force/torque readings as the input of the controller. Vision-based RL approaches have also been studied. For example, [36] proposes using residual RL, to correct a nominal policy produced by state-of-the-art off-policy algorithms to perform connector insertions. Finally, we also find multi-modal sensing approaches. In [37] authors study the importance of each modality (vision, F/T, and positional sensing) for peg insertion tasks in simulation which then transfer the policy into the real system. The authors of [38] proposed empirical studies on the effects of the three different modalities for insertion tasks in a real robotic set-up using behavioral cloning.

Most RL approaches above do not use force/torque or tactile signals to extract geometric information of contacts, but rather for detecting in/out of contact. In [26] the authors demonstrate the potential use of vision-based tactile sensors by explicitly estimating the contact error with supervised learning, and driving the insertion with a proportional controller. In this work, we directly train an RL policy to guide the insertion with tactile feedback from 2 vision-based tactile sensors in a more challenging insertion scenario with different objects and environments.

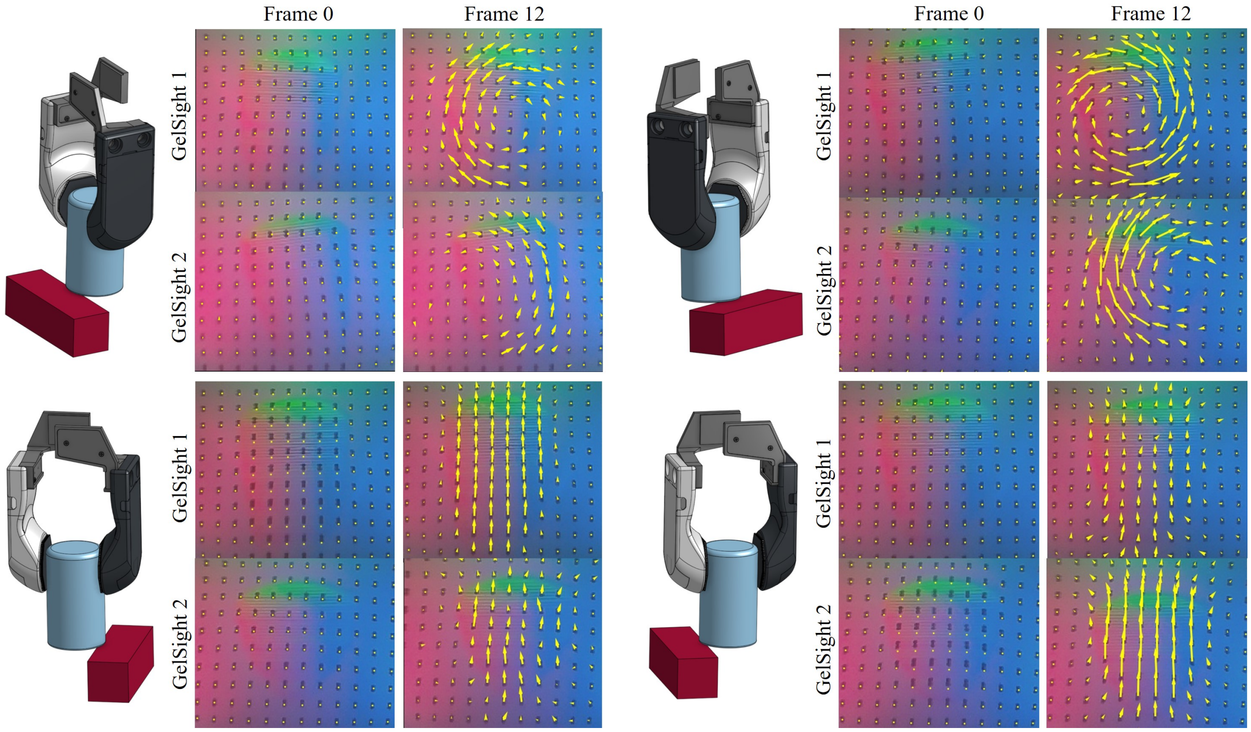


Figure 2.2: Tactile signals for four contact configurations. In the first 2 cases, the object contacts an edge in the environment that is perpendicular to the plane of the sensors, and consequently, the object rotates in the plane of the sensors. Note that the rotation directions are opposite in cases 1 and 2. In the last 2 cases, the object contacts an edge parallel to the plane of the sensor and consequently rotates out from the sensor plane. Note that the relative vertical displacements in the two sensors are different for cases 3 and 4.

2.3 Methodology

In this section we introduce the method by addressing the following 4 questions: 1) Why is tactile sensing essential for capturing contact with the environment? 2) Why is RL a good fit for the insertion task? 3) Why is curriculum learning needed in the RL training? 4) How do different representations of the tactile signal affect the learning agent?

2.3.1 Tactile Sensing for Contact Localization

Humans routinely do blind insertions by feeling the interaction between an object and the environment through the grasp on the object. Contact with the environment generates small forces and small object displacements which are captured by force and tactile sensors in the fingers and hands. Similarly, we use two high-resolution tactile sensors to capture the subtle rotation signals of the object during contact in the insertion task. In [26], we have shown that it is possible in some configurations to recover explicit external contacts from those subtle rotation signals. These sensors capture a depth image during contact as well as the in-plane displacement of an array of markers on the sensor surface.

Figure 2.2 illustrates the signals generated on the sensors under different contact configurations. In the example, the tactile images are captured during the contact phase between a cylindrical object and an edge in the environment, while the gripper moves down vertically. To visualize the flow of the tactile imprint during that phase, we use yellow arrows to draw the relative displacement of the marker field during that contact phase for Sensor1 (black) and Sensor2 (silver).

The key observation is that some different contact locations generate different patterns of tactile flow. In the first two configurations, the object contacts an edge that is perpendicular to the surface of the tactile sensors. In this case, the object rotates in the plane of the sensors. The captured patterns of rotation contain information about the external contact point. Similarly, in the last two configurations, the object contacts an edge in the environment that is parallel to the surface of the tactile sensors. In this case, the object rotates out from the plane of the sensors. These capture the vertical displacements of the contacts. For simple contact configurations as in Figure 2.2, and if the geometry of the object and environment are known, it is not difficult to directly describe how the tactile flow should look. In general, however, the relationship becomes more involved for complex contact formations. In any case, the tactile image sequence during the contact period from two tactile sensors contains key information about the location of external contacts.

2.3.2 Deep RL Controller

In previous work, Dong and Rodriguez [26] used a supervised learning (SL) model to directly estimate the misalignment error between object and hole from GelSlim [27] tactile feedback in the form of image sequences. While this approach might work well in simple scenarios, due to the partial observability, the SL model may fall apart and make the controller diverge. We will show in Sec. 2.4 that this approach does not work well in more complex environments such as a rectangular hole.

A more general solution is to model the insertion problem as a sequential decision problem and use a reinforcement learning (RL) algorithm to find the optimal policy. The input to the RL framework is the tactile image sequence during contact. The action is the robot motion in operational space. The reward is related not only to the pose error corrected in each step but the success or failure of the insertion. The optimal RL policy tries to achieve the maximum reward by inserting the object inside the hole within a sequence of actions, instead of greedily correcting the pose error in each step. In this way, the policy can potentially perform better under some partially observable states, where it is not possible to estimate the full pose error with the tactile signals. This is the key advantage of the RL policy over an SL policy detailed in Section 2.4.

2.3.3 Curriculum Learning

Due to the challenges of simulating the tactile signal with accuracy during contact, we train the insertion policy on a real robot. In order to improve the data efficiency of RL and make the problem feasible to solve in a reasonable training time, we explore the use of curriculum learning.

We design the learning curriculum by scaling the complexity of the insertion environment. In Figure 2.3, we introduce 4 insertion environments shown in red: line wall → corner wall → U wall → hole. Note that each environment adds one more constraint to the previous environment, and forces the insertion policy to learn to handle that new constraint. Intuitively, the curriculum is asking the RL policy to incrementally learn to: 1) distinguish the contact direction in one axis in the line wall environment; 2) reason about the contact directions in 2 axes in the corner environment; 3) control the step size to avoid overshoot in one axis and rotation in the U environment; and 4) control step size in both axis and rotation in the hole environment.

In order to generalize the RL policy to different objects, we train the policy with 4 objects with different shapes (cylinder, hexagonal cylinder, elliptical cylinder, and cuboid) shown in Figure 2.3. We also use curriculum learning by gradually increasing the number of objects to insert in the line environment. Generally, the cylinder, hexagonal cylinder, and elliptical cylinder are easier to insert, since they are less sensitive to rotational error. The cuboid is the most difficult to insert because a small misalignment in the yaw will block the insertion.

2.3.4 Tactile representation

The choice of the tactile signal representation affects the generalization of the policy to new objects since the combination of a high-dimensional space, and a small number of training objects can easily lead to overfitting. High-resolution tactile images encode rich information on the contact surface. Some information content such as texture might not be relevant to the insertion task, while some other information might be more closely related to the location of external contacts, such as force distribution and tactile flow. Here we study the effect of data representation of the tactile signal on the RL policy using 2 different types of data input: 1) the raw RGB image, and 2) the marker flow image.

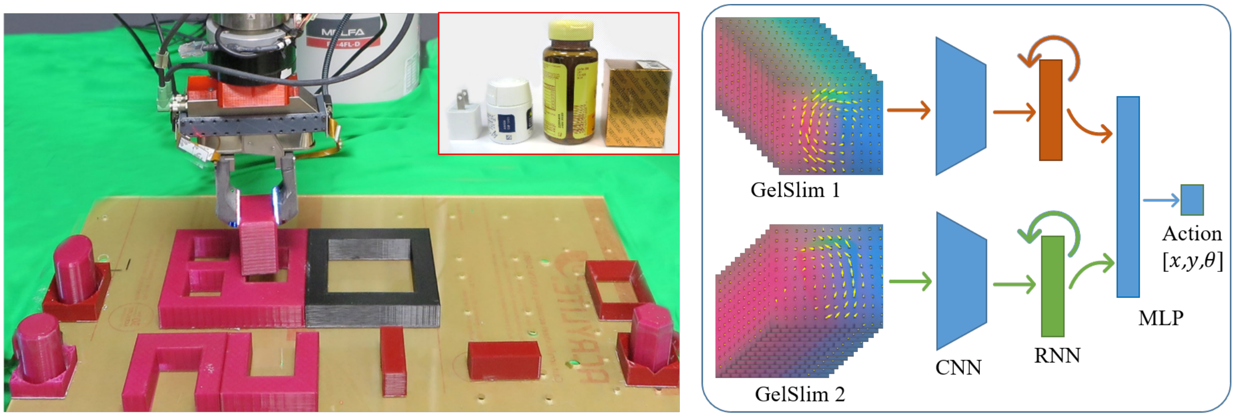


Figure 2.3: Left: the experimental setup including a Mitsubishi Electric RV-4FL model robot arm, a WSG-32 parallel-jaw gripper, 2 revised GelSlim sensors, 4 sets of insertion environments, 4 training objects (cylinder and elliptical cylinder on the left, hexagonal cylinder on the right, cuboid held), and 4 novel everyday objects (phone charger, small bottle, big bottle, and box). Right: The Actor Neural Network model in TD3. The CNN includes 4 convolutional layers with 5, 3, 3, 3 kernel sizes and extracts 1×512 features. The RNN are 2 layers of LSTM with 512 memory units and output $.1 \times 512$ features. The MLP includes 2 fully connected layers with 512 and 256 hidden units. The marker flow is for visualization, not included in the training.

2.4 Experimental Details

2.4.1 Experimental Setup

Figure 2.3 shows the experimental setup, including a 6-DoF Mitsubishi Electric RV-4FL robot arm, a WSG-32 parallel-jaw gripper, two GelSlim sensors [27], one Mitsubishi F/T sensor 1F-FS001-W200 (only used to train the F/T RL policy training), four training objects and their resetting fixtures, and four environments for insertion, including line walls, corner walls, U walls and holes. Note that the fixtures on the sides of the experimental setup are used for automatic object pose resetting for the training objects. Note also that the clearance of the U wall and the hole environments is 3mm, while the average width of the objects is 35mm.

2.4.2 Insertion Experiment

Similar to [26], we assume that during actual operation, a noisy position of the hole would be estimated by a vision system, and the objective of the tactile-RL policy is to correct for the remaining relative pose error. To emulate this scenario, we introduce random translation error ranging from -6 $\sim$ 6 mm in the x and y axis (17% of the object's width) and rotation error in the yaw angle (θ) ranging from -10 $\sim$ 10 degrees. The resetting mechanism for objects, allows to pick them with controlled noise, different grasping forces and different heights. This is helpful to avoid overfitting to the height of the object or certain local tactile image features.

Experimental process For every episode, the robot randomly chooses and grasps an object and moves it to the top of the insertion environment with additional translational and rotational errors. Subsequently, the robot moves down vertically to attempt an insertion, while monitoring collisions with slip detection [39]. If the slip detector is not triggered while moving down, that indicates a successful insertion. Otherwise, the insertion is blocked by the environment. We capture the tactile image sequences (or the F/T data when using the F/T policy) from the two tactile sensors during the contact and feed them to the controller. The robot moves the object to the next position according to the output of the controller and starts another trial until either the object has been inserted or the insertion has failed. The robot resets the pose of the object by returning the object to its fixture after every 10 trials during the training process.

2.4.3 Deep RL Policy

We choose an off-policy RL algorithm, the Twin Delayed DDPG (TD3) [40], to train the tactile-RL policy for better sample efficiency. The underlying RL framework is represented by a tuple of (S, A, p, R, γ). Here we use 12 images (downsampled from 30 images within 0.5 seconds of contact to avoid the gradient vanishing problem in the RNN model, with 640×480

pixel resolution) captured by each tactile sensor (24 in total) during the contact period as the state $s_t \in S$. The terminal states are either the successful insertion state, or the state that has an error larger than a threshold in any axis, or the state when the attempted number of insertions is over a limit. We set the max error E_{max} in the x and y axis to be 12mm, and 15 in θ. The maximum number of attempts is 15. The action $a_t \in A$ is the continuous robot displacement in $(\Delta x, \Delta y, \Delta \theta)$ in the current gripper frame. The reward function R, shown in (1), is calculated with the difference between the current contact error (e_t) and the previous contact error (e_{t-1}). A constant penalty term is given for each insertion attempt (P), and an additional reward (R_s) is given to a successful insertion ($\alpha = 1$, otherwise $\alpha = 0$).

$$R_t = e_{t-1} - e_t - P + \alpha R_s \tag{2.1}$$

We use a discount factor $\gamma = 0.99$ for computing the Q function. As mentioned in Sec. 2.3, we use curriculum learning for RL training, and we stop training when the mean of the rewards crosses a threshold and the standard deviation is lower than a threshold in the last 30 episodes.

Model architecture TD3 is an actor-critic method [40]. We use a CNN+RNN+MLP model for the actor network with the 2 tactile image sequences used as input shown on the right-hand side of Figure 2.3. Since the true state (the contact errors) of the task is known and thanks to precise initial grasps and careful resetting mechanisms during training, we directly use the current contact error in (x, y, θ) axis and the action a_t from the actor model as the input to the critic model, instead of the tactile images. For the critic model, we use a small MLP model to estimate the $Q-$function. This can potentially speed up the convergence of the RL policy.

Training details To further accelerate training of the RL policy, we bootstrap the actor network with a supervised learning policy trained with 300 data points collected under a random policy in the first environment. In the first 50 episodes of RL training, we freeze the actor network to avoid updates by the untrained critic network. This helps to accumulate positive examples into the replay buffer.

2.4.4 Supervised Learning Policy

To compare with previous work [26], we train a supervised learning (SL) policy with the data (tactile image sequence) and labels (object-hole misalignment) saved during the RL policy training as a baseline. The SL policy takes the marker flow image sequence as the input and outputs the estimated contact error in the x, y, and θ axis. For a fair comparison, we use the same architecture for the SL policy as the RL actor model. The SL policy is trained to estimate the contact error with an MSE loss. During execution, we use the action which is the opposite of the output of the SL policy to minimize the contact error iteratively. The maximum number of attempts is also 15.

2.4.5 RL policy with F/T sensor

We train an RL policy with F/T sensing feedback as another baseline. We collect a stream of F/T signals (32 samples for each of the 6 dimensions of F/T, downsampled from 120) during each contact (0.5 s) as the input to the RL policy. The downsampling is used to avoid the gradient vanishing problem for the RNN model, but still keep the original force profile. We remove the torque about the vertical axis T_z from the input since it only changes minimally during contact. We keep the general RL framework and only modify the neural network architecture of the actor network to a smaller RNN + MLP model to adapt to the data format of the F/T sensor. Since the model size is much smaller, we can train the policy directly in the hole environment with an SL policy initial bootstrap until the policy converges.

Table 2.1: Performance of different policies tested with 4 training objects and 4 novel objects

		Cylinder	Hexagonal cylinder	Elliptical cylinder	Cuboid	Big bottle	Small bottle	Phone charger	Paper box
RL*	Success	97.1%	**97.1 %**	98.0%	**89.6%**	80.8%	**96.7%**	**96.8%**	**70.1%**
	Attempt	2.96	3.83	**2.34**	**5.42**	3.53	**2.10**	**2.55**	**7.91**
SL	Success	84.7%	70.0%	93.7%	15.2%	27.0%	76.3%	42.46%	18.3%
	Attempt	3.04	3.34	2.60	3.83	4.23	2.02	5.10	3.58
RL w/o Curr	Success	99.1%	89.5%	98.0%	57.1%	30.3%	36.23%	71.2%	9.01%
	Attempt	2.97	5.41	4.15	11.3	8.79	11.8	6.69	7.32
RL RGB	Success	98.8%	93.1%	**99.15%**	88.0%	37.8%	95.7%	74.1%	13.4%
	Attempt	2.26	**2.65**	2.42	**5.35**	5.49	3.16	4.92	8.35
RL F/T	Success	**100.0%**	89.3%	**99.15%**	61.2%	**99.5%**	95.2%	53.5%	54.1%
	Attempt	**1.82**	3.26	2.52	5.35	**2.48**	3.82	6.53	8.35

2.5 Results

We evaluate the performance of the policies by conducting 250 insertion experiments under different initial pose errors, with each of the 4 training objects and each of the 4 novel objects shown in Figure 2.3. The initial pose errors are uniformly sampled in the error space, where the errors in x and y axis range from -5mm to 5mm, and the error in θ is from $-10°$ to $10°$. The performance metric includes the success rate and the average number of attempts, which is averaged only over the successful cases.

2.5.1 RL Policy with Curriculum and Tactile Flow (RL* policy)

Convergence of the RL* policy is surprisingly fast. Due to the warm start of the actor network, the policy can already insert the cylinder most of the time in the line-wall environment at the start. Since some object-specific features (texture and shape) of individual objects are removed in the marker flow, the policy trained with 1 object can quickly adapt to the other three objects. Because the policy from the first environment can already perform well in the corner-wall environment, we stop training after 25 episodes (50 data points, which means 2 attempts for each episode). The U-shaped environment is much harder compared to the previous 2 environments, and the policy needs around 150 episodes (500 data points) to adapt. The policy converges after 300 episodes (2000 data points) in the hole environment, with the warm start from the previous environment. From our observation, the policy can quickly learn how to insert the first three objects but spends most of the time learning how to correct the rotation error for the cuboid object. The whole training process takes around 8 hours, with 3000 insertion attempts in total.

The **performance** of the RL* policy in the test experiments is illustrated in Table 2.1. For the 4 training objects, the RL* policy achieves a 90% success rate with the cuboid and over 97% success rate for the remaining 3 objects. The cuboid object needs around $5 \sim 6$ attempts to be inserted, whereas the other three objects only need $2 \sim 3$ attempts. This result agrees with the respective difficulty of the tasks. As we described earlier, whereas the policy needs to correct the rotation error of the cuboid to complete insertion, the other three objects are not sensitive to rotational error. Misalignment in the rotation is hard to observe, especially when accompanied by errors in the x and y axis. This is because the tactile signal is generated by a complex combination of the object's rotation into and rotation parallel to the sensor surface. The policy can also generalize well to 4 novel objects. It achieves over 96% success rate with the small bottle and the phone charger. Both of them are either round or have round corners, which makes the insertion easier. The success rate of the big bottle drops to 80% because the chamfer on the bottom edge changes the way the object rotates. The box can be inserted in over 70% of the tests, which has the lowest success rate but is also the most difficult task. Compared to the training object cuboid, the success rate is lower but the RL* policy still achieves good performance. The novel object and training objects require a similar number of attempts.

Most of the **failure** cases happen when both the translational (in x and/or y) and rotational components (in θ) of the initial pose error are large. Since the rotation of the object is too little to be captured by the tactile sensors in these cases, the RL policy can output an action in the opposite direction, resulting in an even bigger error, until it finally diverges.

2.5.2 Baseline Comparison

SL policy According to Table 2.1, the SL policy performs a bit worse in the first 3 training objects, compared to the RL* policy, but can barely insert the cuboid during the test. In the test cases with 4 novel objects, only the small bottle is inserted with over 70% success rate, and the others have significantly lower success rate. The RL* policy is also more efficient as to the number of attempts. We find that the cuboid objects are particularly challenging for the SL policy, due to the fact that the tactile signals are similar in many contact configurations, and it's very difficult for the SL policy to make the right move. In addition, considering the SL policy is trained with MSE loss, a safe choice for the controller is to output 0 when the signal is not distinctive, which makes the SL policy get stuck. This comparison demonstrates that the RL policy learns a better strategy to insert objects even with some partially observable states.

RL policy without curriculum To evaluate the improvement of the curriculum training on the convergence speed, we train another RL policy directly in the hole environment with all 4 training objects. To keep it consistent with the RL* policy, we also bootstrap the RL policy with the same SL policy and use maker flow as the input. We stop the training when it reaches the same amount of data used for RL* policy training. The performance of this policy is shown in the 4th row of Table 2.1. The policy achieves a similar success rate for the first three objects but uses nearly twice the number of attempts compared to RL*. The policy fails in almost all of the cuboid insertions with large rotational error, and only gets a 57% success rate, using 11 attempts on an average. Based on the comparison, we can interpret that 1) localizing the contact direction in the x and y axes is easy to learn, 2) optimizing the number of attempts, relating to precisely localizing the amplitude of the contact error and a high-level insertion strategy to quickly gather information, is harder, 3) estimating the rotational error is also difficult and data-intensive. This RL policy does not generalize to the new objects well and only achieves a 30% average success rate.

RL policy with RGB input The raw RGB tactile image contains task-relevant as well as detailed object-specific features. It is much easier for the RL policy to use RGB images to overfit the training objects. This hypothesis agrees well with the result shown in Table 2.1, 5th row. The performance of the policy with training objects is as good as that of the RL* policy. However, the success rate drops dramatically in the test of novel objects, especially with the big bottle and paper box. The policy keeps outputting the same action during the testing of these two objects, which reveals that the policy is likely overfitted. Using a proper tactile representation is important for generalization. Marker flow may not be the best representation, and we leave the problem of learning a good representation for future work.

RL policy with F/T sensing achieves almost 100% success rate with cylindrical objects with round edges in both the training (cylinder and elliptical cylinder) and novel objects (2 bottles), shown in Table 2.1, 6th row. In some of the objects, it uses even fewer attempts compared to the RL* policy. However, its success rate drops to 89% for hexagonal ellipse and further drops to around 55% for the three cuboid shaped objects (cuboid, phone charger, paper box). According to our observation, the policy with the F/T sensor is good at localizing contact errors in x and y directions but has trouble distinguishing the rotation error. Compared to the F/T sensor, the tactile sensor observes not only the force information but also the object motion in the contact phase, which may contain useful information to differentiate the rotation error.

2.6 Conclusion

In this chapter, we proposed an RL insertion policy with tactile feedback from 2 GelSlim [27] tactile sensors without prior knowledge of the object's geometry. We show that our proposed RL policy with designed curriculum training and tactile flow representation, provides several advantages over other tactile-based baseline policies, including: 1) supervised learning policy,

2) RL policy without curriculum, 3) RL policy with RGB input. Furthermore, we also showed the advantage of using tactile over F/T sensors in terms of generalization, which are the most widely used sensors for these kinds of applications.

Chapter 3

Active Extrinsic Contact Sensing:

How Can We Exploit Tactile Sensing More Efficiently?

We propose a method that actively estimates contact location between a grasped rigid object and its environment and uses this as input to a peg-in-hole insertion policy. An estimation model and an active tactile feedback controller work collaboratively to estimate the external contacts accurately. The controller helps the estimation model get a better estimate by regulating a consistent contact mode. The better estimation makes it easier for the controller to regulate the contact. We then train an object-agnostic insertion policy that learns to use the series of contact estimates to guide the insertion of an unseen peg into a hole. In contrast with previous works that learn a policy directly from tactile signals, since this policy is in contact configuration space, it can be learned directly in simulation. Lastly, we demonstrate and evaluate the active extrinsic contact line estimation and the trained insertion policy together in a real experiment. We show that the proposed method inserts various-shaped test objects with higher success rates and fewer insertion attempts than previous work with end-to-end approaches. See supplementary video and results at https://sites.google.com/view/active-extrinsic-contact.

3.1 Introduction

Sensing and utilizing tactile feedback between fingers and grasped objects is key to dexterous manipulation skills [41]. Tactile sensing can be used as a feedback signal to regulate desired contact configurations [42]. One major problem in tactile feedback control is localizing and controlling an external contact between the grasped object and its environment. For example, consider pivoting an unknown object resting on a surface while avoiding slipping at the contact between the object and the surface. It requires an ability to regulate the external contact using indirect observations, possibly through tactile sensing.

In the peg-in-hole insertion task, the external contact matters. During the insertion attempt, a misalignment between the peg and the hole leads to unintended contact. The contact triggers a tactile signal, which can be used to localize the contact or plan the next insertion attempt. Key challenges in this task are as follows:

- The tactile signal is a partial observation of the contact state; many different contact configurations can cause the same tactile signal [32].
- The frictional contact mechanics that govern the alignment and insertion dynamics are sensitive to switching contact modes.

This work tackles both the extrinsic contact localization problem and the peg-in-hole insertion task in a combined framework. For the extrinsic contact localization, we use a factor graph solved with incremental smoothing and mapping (iSAM) [43] to fuse the information of robot proprioception and tactile measurements. The factor graph works collaboratively with an active tactile feedback controller that attempts to pivot the object about a regulated external contact to generate sufficient observations to estimate the contact line between an unknown object and an unknown environment. The contact line estimation is then used as an input to the insertion policy, as opposed to the end-to-end approach where the policy takes directly as input the raw tactile feedback. Since the input to the policy is a low

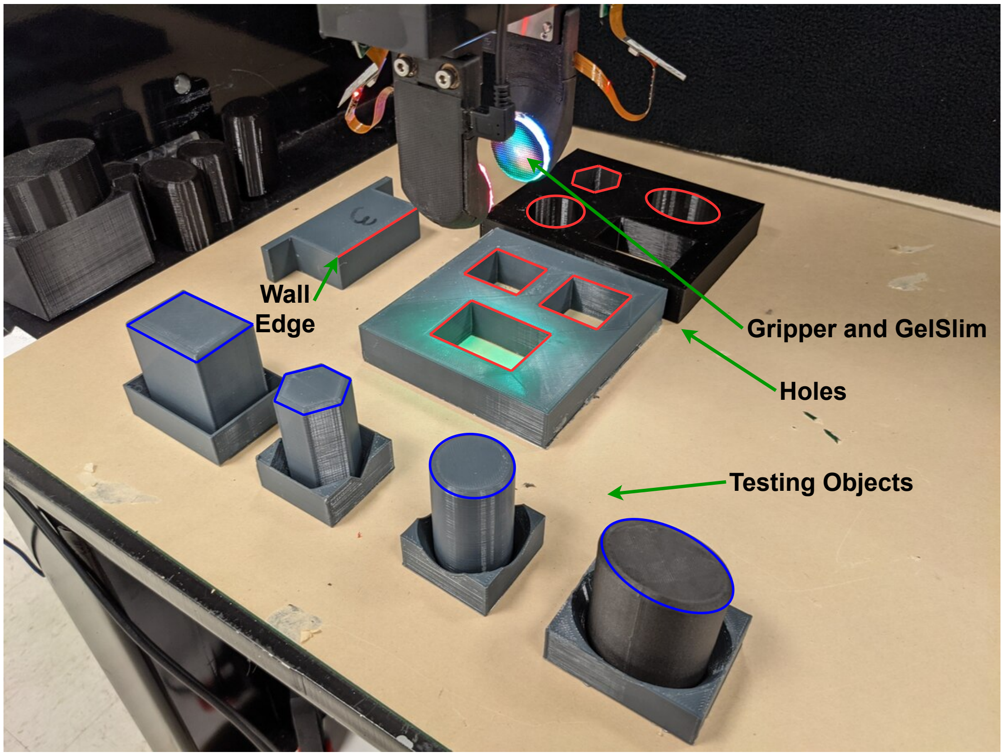


Figure 3.1: Experimental setup: Gripper with GelSlim [27], different objects (blue), and different environments (red)

dimensional representation (a contact line), the policy training can be done in a simple 2D geometric simulation, so there is no need to collect training data in a real experiment. Lastly, we demonstrate and evaluate the extrinsic contact line estimation and the trained insertion policy in a set of real experiments.

3.2 Related Work

3.2.1 Tactile Sensing and Feedback

Prior work showed that tactile measurements could be used for state estimation. Bicchi, et al. [22] used force measurements to estimate a contact location when the force is exerted on the robot with known geometry. Yu and Rodriguez [44] combined force and visual sensing to estimate the pose of a known planar object in planar manipulation. They also used a similar framework to estimate the SE(3) pose of a known object in touch with an external environment [21].

Vision-based tactile sensors like GelSight [14] and GelSlim [27], [45] enabled accurate estimation of contact states. These sensors capture the physical interaction between the robot fingers and the grasped object as high-resolution images. Bauza, et al. [46], [47] used tactile images to map and localize the relative pose of a known grasped object. Ma, et al. [48] conducted inverse finite element method (iFEM) on tactile images to reconstruct a dense tactile force distribution. Sodhi, et al. [49] developed a factor graph model that finds a maximum a posteriori (MAP) estimate of an end-effector and an object pose in planar manipulation with a vision-based tactile sensor [50]. While the work above focused on estimating the states of an object or an interface in direct touch with a tactile sensor, Ma, et al. [25] focused on localizing extrinsic contacts. They accumulated a sequence of tactile images and used least-squares fitting [51] to estimate the location of the extrinsic contact. However, all these estimation methods are passive, so they are limited to using the information they are presented with.

The tactile sensing can be used as a feedback signal to regulate the desired contact configuration. Dong, et al. [52] introduced a module that monitors incipient slip on the tactile sensor and uses it to maintain a stable grasp. Hogan, et al. [53] developed a closed-loop tactile controller for dexterous manipulation primitives. She, et al. [54] showed tactile sensing for manipulating a cable. They learned a linear model for the cable sliding dynamics and implemented a linear quadratic regulator (LQR) to keep the cable near the sensor center while sliding through the cable.

3.2.2 Peg-in-Hole Insertion

Early studies on peg-in-hole insertion relied on passive compliance of the gripper [28]. Other works assume a known object-hole model [30], [31]. These methods are object-specific and difficult to apply to unknown objects. More recently, model-free, end-to-end, learning-based

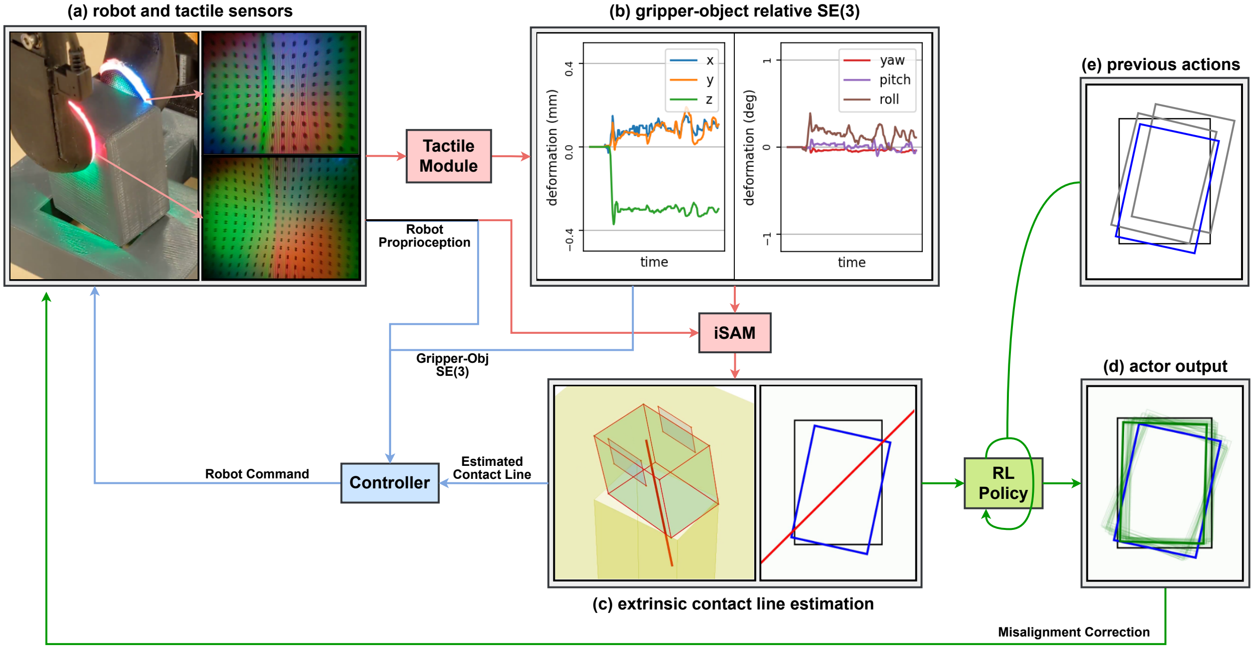


Figure 3.2: Approach Overview. Estimation module (red) and active tactile feedback controller (blue) run collaboratively to estimate extrinsic contact line. RL policy (green) takes the estimated extrinsic contact line as input and computes the next action. (a) Insertion attempt and tactile images captured by GelSlim fingers. (b) Gripper-Object relative displacement computed by a learned tactile module. (c) 3D and top view of the extrinsic contact line estimation. The bold red line is the current estimate. (d) RL Actor output. The blue rectangle is the current pose and the green rectangles are the candidate poses for the next insertion attempt. (e) History of previous attempts that feed into the recurrent RL architecture.

approaches have been proposed [18], [26], [55], [56]. [55] used deep reinforcement learning (RL) for a high precision peg-in-hole task. They used a force/torque (F/T) sensor measurement and a robot position as input to a discrete action policy. [56] fused force and vision to address the peg-in-hole insertion task. [26] used supervised learning to map a tactile image sequence to a misalignment between peg and hole. However, since the tactile image does not fully observe the state, their approach showed sub-optimal performance. The same group of researchers used end-to-end RL to overcome this limitation and showed a performance improvement [18]. Simulating raw images is difficult in these end-to-end approaches, so the training data was collected in real experiments.

3.3 Methodology

The task we solve is a typical peg-in-hole problem. We make several assumptions to implement our framework:

- Object bottom surfaces and hole top surfaces are flat.
- The misalignment between object and hole is an SE(2) displacement in the plane of contact.
- Objects and holes are un-chamfered.

The approach comprises two parts: the active extrinsic contact sensing and the insertion policy. In the active extrinsic contact sensing, a factor graph estimator works collaboratively with an active tactile feedback controller to estimate the extrinsic contact line. The controller helps the factor graph improve the estimation by regulating a consistent extrinsic contact mode. The improved estimation from the factor graph helps the controller better regulate the extrinsic contact. Then, the insertion policy learned in simulation with reinforcement learning (RL) takes the estimated contact line as input and computes the next action for the insertion.

3.3.1 Tactile Module and Factor Graph

We use GelSlim 3.0 [27], a vision-based tactile sensor, to capture the deformation image on the robot finger during the insertion (Figure 3.2a). The image is passed to a convolutional neural network (CNN) architecture, the tactile module, trained with supervised learning to estimate the relative SE(3) displacement between the gripper and the object due to the compliance of the finger (Figure 3.2b). The gripper-object relative displacement and the robot proprioception data are then used in the factor graph to infer the extrinsic contact line (Figure 3.2c).

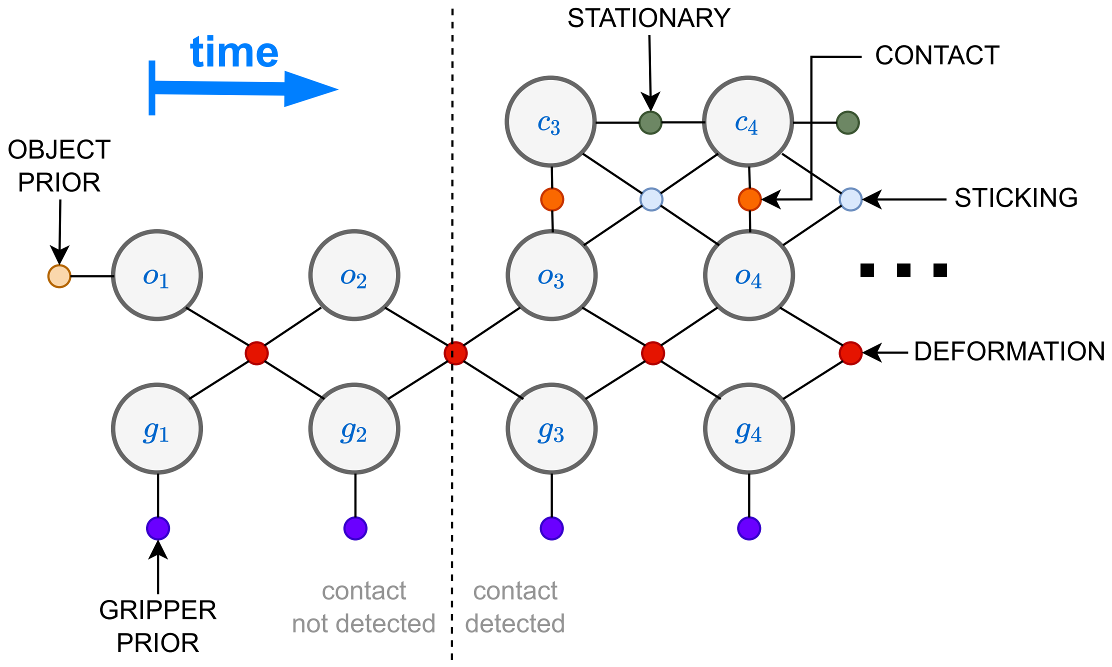


Figure 3.3: Estimation factor graph

Figure 3.3 shows the factor graph. Each color of small circles represents different types of factors. g_i, o_i, and c_i are the SE(3) gripper pose, object pose, and contact line as shown in Figure 3.4 ; xy surface of o_i represents the bottom surface of the object and x-axis of c_i represents the estimated contact line. iSAM solver computes a maximum a posteriori (MAP) estimate for trajectories of the gripper pose G, the object pose O, and the contact line C, given the gripper pose measurements based on robot proprioception R, the gripper-object relative displacement D, and the initial object pose prior b_o:

$$G^*, O^*, C^* = \underset{G,O,C}{\arg\max}\, P(G, O, C, R, D, b_o) \tag{3.1}$$

Assuming Gaussian noise, solving for the MAP estimate becomes a nonlinear least-squares

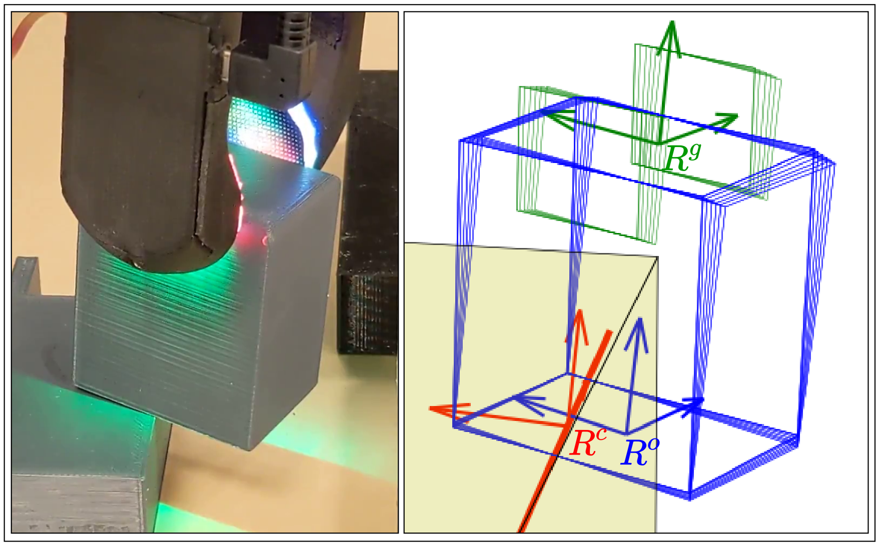


Figure 3.4: Left: An object in contact with an edge. Right: Gripper (green) and object (blue) trajectories and an estimated contact line (red).

problem:

$$
\begin{aligned}
G^*, O^*, C^* = \underset{G,O,C}{\arg\min} \{ \sum_{t=1}^{T} \{ & ||F_{gp}(g_t, r_t)||^2_{\sum_{gp}} \\
& + ||F_{def}(g_{t-1}, g_t, o_{t-1}, o_t, d_t)||^2_{\sum_{def}} + ||F_{ct}(o_t, c_t)||^2_{\sum_{ct}} \\
& + ||F_{tic}(o_{t-1}, o_t, c_{t-1}, c_t)||^2_{\sum_{tic}} + ||F_{stat}(c_{t-1}, c_t)||^2_{\sum_{stat}} \} \\
& + ||F_{op}(o_1, b_o)||^2_{\sum_{op}} \},
\end{aligned} \tag{3.2}
$$

where $F(\cdot)$ are cost functions for each factor. $||\cdot||_{\sum}$ is the Mahalanobis distance with covariance $\sum$. iSAM enables to add new measurements incrementally and update the estimation in real-time rather than solving it from scratch at every step.

Gripper prior (F_{gp}) and Object prior (F_{op}): We use unary factors to model the uncertainty of the gripper pose and initial object pose:

$$
||F_{gp}(g_t, r_t)||^2_{\sum_{gp}} := ||g_t^{-1} r_t||^2_{\sum_{gp}} \tag{3.3}
$$

$$
||F_{op}(o_1, b_o)||^2_{\sum_{op}} := ||o_1^{-1} b_o||^2_{\sum_{op}}, \tag{3.4}
$$

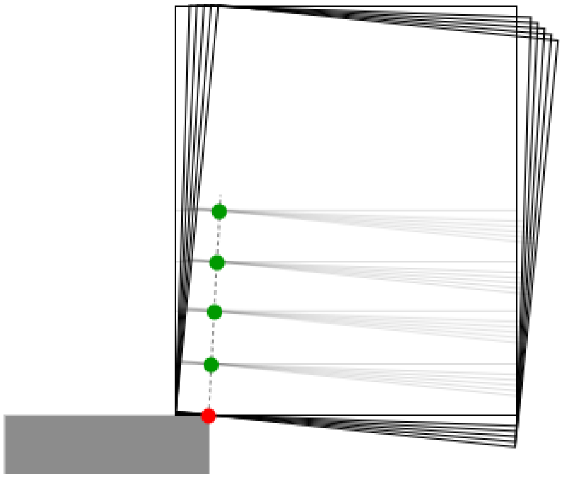


Figure 3.5: An object in touch with an environment. The sticking factor enables the factor graph to distinguish red from green points.

where r_t is the measured gripper pose based on robot proprioception and b_o is the prior knowledge about the initial object pose.

GelSlim deformation factor (F_{def}): Relative SE(3) displacement between gripper and object computed by the tactile module is incorporated as the GelSlim deformation factor:

$$\begin{aligned} &||F_{def}(g_{t-1},g_t,o_{t-1},o_t,d_t)||^2_{\sum_{def}} \\ &\qquad := ||((g_{t-1}^{-1}o_{t-1})^{-1}(g_t^{-1}o_t))^{-1}d_t||^2_{\sum_{def}}, \end{aligned} \tag{3.5}$$

where d_t is the change in the relative displacement from $t-1$ to t.

Contact factor (F_{ct}): To constrain the estimated contact line to lie on the object bottom surface, we use a binary factor:

$$||F_{ct}(o_t,c_t)||^2_{\sum_{ct}} := ||(o_t^{-1}c_t)_{z^c,R_{x^c},R_{y^c}}||^2_{\sum_{ct}}, \tag{3.6}$$

where the subscript z^c, R_{x^c}, R_{y^c} indicates that we only constrain the components that move out of the object's bottom surface.

Sticking factor (F_{tic}): Figure 3.5 shows an example of an object touching an environment and tilting at a small angle. In such a case, it is difficult to infer the actual contact point with only geometric constraints. All the green points in the figure will seem to satisfy the geometric constraints for a certain object height. Therefore, while we can get a reasonable estimate in the horizontal direction, the estimated height of the contact will be uncertain. However, if we use a controller to regulate the external contact to be sticking, only the red dot in Figure 3.5 is now compatible with the constraints. We incorporate it in the factor graph and get a better estimate in the vertical direction. We use a binary factor to impose the sticking constraint:

$$||F_{tic}(o_{t-1}, o_t, c_{t-1}, c_t)||^2_{\sum_{tic}} := ||((o_{t-1}^{-1}c_{t-1})^{-1}(o_t^{-1}c_t))_{y^c}||^2_{\sum_{tic}}, \tag{3.7}$$

where the subscript y^c is the direction on the object's bottom surface perpendicular to the contact line.

Stationary factor (F_{stat}): We assume contact configuration does not change during the motion and the contact line is stationary. This is incorporated by using a binary factor between consecutive timesteps:

$$||F_{stat}(c_{t-1}, c_t)||^2_{\sum_{stat}} := ||(c_{t-1}^{-1}c_t)_{y^c,z^c,R_{y^c},R_{z^c}}||^2_{\sum_{stat}} \tag{3.8}$$

3.3.2 Active tactile feedback Controller

Figure 3.6 shows the flowchart of the active tactile feedback controller. It comprises a push-down phase where the robot moves down until detecting contact, and an active exploration phase where the robot tries to rock and pivot the object about the external contact without sliding. During the push-down phase, it executes a proportional control on the gripper-object relative pose estimated by the tactile module:

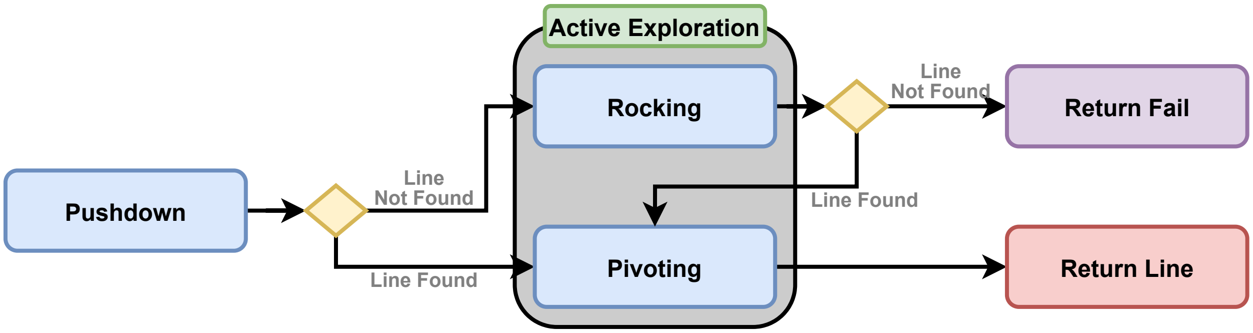


Figure 3.6: Active tactile feedback controller

$$v_z^g = -K_{p,z}(\Delta z_d^g - \Delta z^g) \qquad \text{(vertical)} \tag{3.9}$$

$$\omega_x^g = -K_{p,\phi}(\Delta \phi_d^g - \Delta \phi^g) \qquad \text{(roll)} \tag{3.10}$$

$$\omega_y^g = -K_{p,\theta}(\Delta \theta_d^g - \Delta \theta^g) \qquad \text{(pitch)}, \tag{3.11}$$

where $[\omega_x^g, \omega_y^g, \omega_z^g, v_x^g, v_y^g, v_z^g]^T = [\omega^g, \mathbf{v}^g]^T$ is the body twist of the gripper frame and the Δ's are the relative displacement from when the object was not in contact with the environment. The subscript d is for the desired relative displacement. Δz_d^g is set to a non-zero value to ensure the object contacts the environment with sufficient normal force. $\Delta \phi_d^g$ and $\Delta \theta_d^g$ are set to zero.

If the object does not tilt enough so the factor graph fails to estimate a contact line with enough confidence, it enters the rocking phase. In the rocking phase, the robot follows a cone-like trajectory by setting the desired relative displacement as below:

$$(\Delta \phi_d^g, \Delta \theta_d^g)(t) = \Delta \phi_0^g(\cos \omega t, \sin \omega t)$$

If the factor graph fails to find a contact line even after the rocking, the controller stops and the estimator returns a failure signal.

If the factor graph succeeds in estimating an extrinsic contact line either in push-down or rocking, the controller enters the pivoting phase to help the factor graph to get more accurate estimate. The controller tries to pivot the object around the extrinsic contact line while avoiding slipping and maintaining a constant tactile deformation. It is assumed that the tactile deformation will remain constant during the pivoting if the object pivots without slipping and maintains a constant contact force. Proportional control is used to maintain the deformation as constant as possible:

$$\omega^g = \pm\omega_0 \hat{x}^{gc} - K_{p,\beta}(\Delta\beta_d^g - \Delta\beta^g)\hat{y}^{gc}, \tag{3.12}$$

$$\begin{aligned}\mathbf{v}^g = \mathbf{p}^{gc} \times \omega^g - K_{p,\alpha}(\Delta\alpha_d^g - \Delta\alpha^g)\hat{y}^{gc} \\ - K_{p,z}(\Delta z_d^g - \Delta z^g)\hat{z}^g,\end{aligned} \tag{3.13}$$

where $\hat{x}^{gc}$ and $\hat{y}^{gc}$ are the $\hat{x}^c$ and $\hat{x}^c$ in the gripper coordinate. $\Delta\alpha^g$ and $\Delta\beta^g$ are the components of the relative rotation change in the $\hat{x}^c$ and $\hat{y}^c$ direction. ω_0 is the rotational speed of the pivoting. $\mathbf{p}^{gc}$ is the origin of the contact line from the gripper coordinate. The desired relative displacement (Δ_d) is updated to a current relative displacement whenever the pivoting direction changes. The first terms of Eq.3.12 and Eq.3.13 rotate the gripper around the current contact line estimate and alter direction ($+\omega_0$ and $-\omega_0$) as it pivots the object back and forth. The second term of Eq.3.12 and the third term of Eq.3.13 ensure the object is being pushed down while maintaining the same line contact. The second term of Eq.3.13 translates the gripper to the direction it reduces slipping.

A key idea in the above method is the synergistic interaction between the controller and the estimator. A better contact line estimation helps the controller pivot the object with less slipping. On the other hand, better pivoting around a consistent axis allows the estimator to get a more accurate contact line estimate.

3.3.3 RL Policy

The RL policy takes the estimated extrinsic contact line from the estimation graph as the input and computes the SE(2) pose correction for the next insertion attempt (Figure 3.2d). Since the input to the policy is a low dimensional representation (a single contact line), it is computationally trivial to simulate the policy; given two random shape polygons, each representing the object and the hole, and an SE(2) misalignment between the two, it is easy to find a contact line that the object can pivot around as can be seen in Figure 3.2c(right). Further details are discussed in Section 3.4.

3.4 Experimental Details

3.4.1 Experimental Setup

Figure 3.1 shows the experimental setup. We use a 6-DoF ABB 120 robot arm and a WSG-50 parallel jaw gripper. GelSlim 3.0 [27] sensors are mounted on each side of the jaw gripper. Four types of 3-D printed testing objects are used on seven types of holes and one single wall environment. The average width of the objects is 35mm and the clearance between the objects and the holes is 2.25mm. The top view of each object-hole pair is shown in Table 3.2. We use Pytorch [57] for training the tactile module and the RL policy, and iSAM2 solver [58] implemented in the GTSAM library [59] for the factor graph optimization.

3.4.2 Tactile Module Training

The tactile module is trained with supervised learning to estimate the gripper-object relative displacement. It takes a pair of reference tactile images from each side of the sensors at one time and a pair of query tactile images at another time then computes the relative displacement change between the two times. We fix the four objects then grasp them with random pose and force. Then we randomly wiggle the gripper with Ornstein-Uhlenbeck process [60] to collect a training sequence of tactile images. The wiggling motion is clipped with a max limit of (0.25mm, 0.5mm, 0.5mm, 1.2°, 0.6° 0.2°) in the $(x, y, z$, roll, pitch, yaw) direction. We regrasp the object after every 10 seconds. 100 sequences for each object are collected and combined in one large training set. The trained module estimates the relative displacement in the $(y, z$, roll, pitch, yaw) direction with reasonable accuracy: RMSE of (0.07mm, 0.06mm, 0.2°, 0.1°, 0.05°). However, it showed less accuracy in the x direction, the direction perpendicular to the sensor surface: RMSE of 0.12mm. This is expected since it is the direction where the finger gel skin affords less deformation.

3.4.3 Active Extrinsic Contact Sensing Experiment

Before testing the entire framework in the insertion task, we decouple only the active extrinsic contact sensing part and test the estimation accuracy on the single wall environment. For every grasp, we randomize the grasping height and force. We also vary the horizontal translation and horizontal rotation misalignment between the object and the wall. We say the misalignment is zero when the wall's edge and the object's x-axis match when seen from the top view. Translational error is uniformly sampled from $-12 \sim 12$ mm and rotational error is sampled from $-90° \sim 90°$.

We compare the performance with some ablation models as seen in Table 3.1. Note that 'w/o Deformation Factor' means that we do not incorporate deformation information into the factor graph but we still use it as the feedback signal. 'w/o Control' means that there is no feedback so the gripper pushes down with no tilting and the contact line estimation will only rely on the passive compliance of the GelSlim, as in the case of the earlier work that studied extrinsic contact sensing [25].

3.4.4 RL Policy Training

RL Policy Training is done in simulation. In every episode, to make the policy generalizable to various shapes, we randomize the object polygon by randomly scattering points and

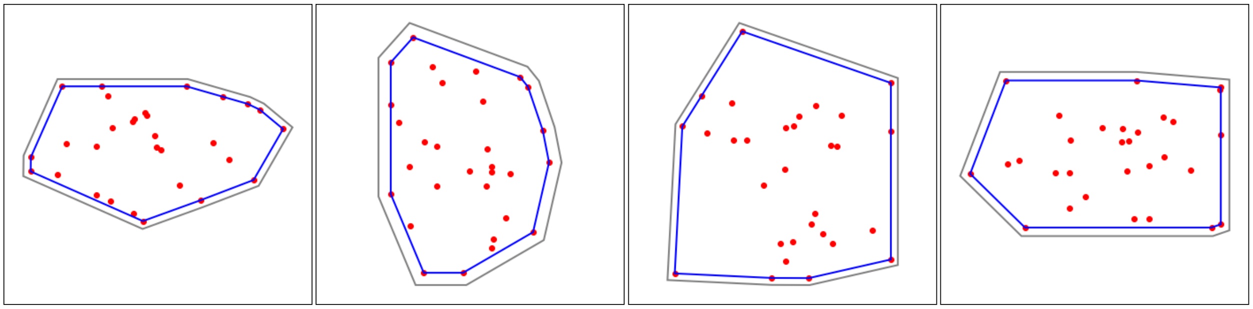


Figure 3.7: Random polygonal object-holes

drawing their convex hull (blue polygon in Figure 3.7). An offset polygon with 2.25mm clearance is used as the hole polygon (black polygon in Figure 3.7). The initial misalignment is sampled from $\pm(12\text{mm}, 12\text{mm}, 15^\circ)$. The simulator takes the object-hole polygon and the misalignment as input. Small Gaussian noise (0.2mm, 0.4°) is added to the misalignment to make the policy more robust. The simulator then computes a valid contact line by connecting intersecting points, if one exists, then a Gaussian noise (4mm, 4°) is added to it and returned as an output. If the object polygon intersects with the hole polygon but there is no valid contact line, the simulator returns no line. Lastly, if the object polygon lies inside the hole polygon without intersecting, the simulator returns an insertion success signal.

As a training algorithm, we use twin delayed deep deterministic policy gradient (TD3 [40]) with a recurrent actor. We formulate the task as a partially observed Markov decision process (POMDP). The observation is the estimated contact line and on which side the object tilts around the contact line. The action is the SE(2) robot displacement from the initial pose. We use the same reward function as in [18]:

$$R_t = e_{t-1} - e_t - P + \chi R_s, \tag{3.14}$$

where $(e_{t-1} - e_t)$ is a decrease in misalignment from the previous step, P is a small constant penalty term, R_s is a success reward, and χ is the success signal (1 if inserted, 0 otherwise). The maximum sequence length for each episode is set to 15. We train the policy for 15,000 episodes, which takes approximately 15 minutes on GeForce RTX 2080, while it is equivalent to about 1,000 hours of uninterrupted robot experiments.

We use 3 layers of long short-term memory (LSTM, [61]) with 64 nodes for the actor network to deal with partial observability. The LSTM alleviates the partial observability by considering the observations and actions of the current and previous steps when computing the following action. We use a multi-layer perceptron (MLP) with 64, 64, and 32 nodes in each layer for the critic. The critic takes the ground truth misalignment and the subsequent action as input and outputs the Q value estimate.

3.4.5 Insertion Experiment

We test the entire framework (Contact Sensing + RL Insertion) with the real test objects and holes. For every episode, we vary grasping height and force. Initial misalignment is sampled from $\pm(12\text{mm}, 12\text{mm}, 15°)$. We evaluate the success rate and the average attempt number. The performance is compared with the same ablation models as in Section 3.4.3.

3.5 Results

3.5.1 Active Extrinsic Contact Sensing Experiment

Table 3.1 shows the performance of the contact sensing experiment on the wall environment in 100 trials per object. Accuracy is the rate at which the method estimates the contact line with horizontal translation error less than 7mm and rotational error less than 25°. We show accuracy separately for easy cases and difficult cases. The easy cases are when the gripper center seen from the top view lies on the same side of the tilting side and the horizontal distance between the gripper center and the contact line is larger than 5mm.

Our method showed an accuracy of 95% for easy cases and 76% for difficult cases. As explained in Section 3.3.1, a model without the sticking factor showed reasonable performance in the horizontal direction but the vertical error was about 3.4 times larger than the original

Table 3.1: Accuracy and average error of the active extrinsic contact sensing

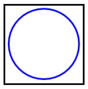
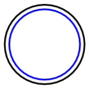
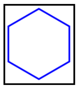
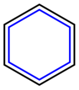
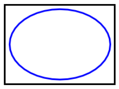
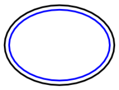
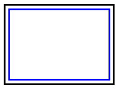
	Accuracy (%)		Average Error		
	Easy	Difficult	Horizontal Translation (mm)	Horizontal Rotation (deg)	Vertical Translation (mm)
Active iSAM	95	76	1.80	5.40	3.31
w/o Sticking Factor	91	79	2.13	5.00	11.2
w/o Deformation Factor	92	74	1.74	5.81	3.08
w/o Deformation Factor and w/o Pivoting Motion	79	55	2.58	6.96	5.50
w/o Pivoting Motion	90	75	2.97	6.57	6.50
w/o Active Motion (rocking & pivoting)	88	34	2.37	6.89	5.36
w/o Control (straight push-down)	28	14	3.70	13.1	5.73

version.

'w/o Deformation Factor' shows similar performance as the original. This is because the deformation is kept almost constant during the pivoting motion so the deformation factor becomes less important. However, 'w/o Deformation Factor and w/o Pivoting Motion' showed poorer performance than 'w/o Pivoting Motion' because the deformation fluctuates significantly during the push-down and the rocking motion.

For 'w/o Active Motion', the accuracy drops, especially for the difficult cases. In difficult cases, the gripper center seen from the top view lies on the wall so it is difficult to tilt with only the push-down motion. In such a case, the active rocking motion is necessary to get a significant tilting angle and estimate a contact line. 'w/o Control' shows the poorest performance meaning that the GelSlim compliance solely does not provide sufficient information to estimate a contact line.

Table 3.2: Success Rate and average number of insertion attempts (in parentheses) for various object-hole pairs.

Object (blue) & Hole (black) Shape							
Active iSAM-RL	100% (1.94)	97% (1.98)	99% (2.40)	100% (2.94)	95% (2.61)	95% (3.04)	97% (4.13)
w/o Sticking Factor	100% (1.92)	100% (2.02)	100% (2.82)	100% (2.70)	96% (3.15)	94% (2.47)	94% (3.64)
w/o Deformation Factor	100% (2.50)	94% (2.32)	98% (2.75)	96% (3.25)	92% (2.83)	96% (2.77)	90% (4.76)
w/o Active Motion	96% (2.15)	94% (2.00)	94% (2.70)	94% (3.51)	90% (3.02)	90% (2.73)	86% (4.77)
w/o Control (straight push-down)	46% (4.17)	46% (5.87)	54% (5.52)	34% (4.82)	58% (4.62)	54% (4.19)	30% (5.60)
RL-end2end (Dong [18])	97% (2.96)	-	97% (3.83)	-	98% (2.34)	-	90% (5.42)
SL-end2end (Dong [26])	85% (3.04)	-	70% (3.34)	-	94% (2.60)	-	15% (3.83)

3.5.2 Insertion Experiments

Table 3.2 shows the insertion performance of the proposed method, ablation models, and previous work [18], [26], in 100 episodes per case. The performance is evaluated in success rate and the average number of attempts until the insertion succeeds. Note that [18], [26] used the shown cases as training cases, while ours used them as testing cases. Also, they used smaller misalignment (6mm, 6mm, 10°) than ours (12mm, 12mm, 15°).

Our method showed a higher than 95% success rate in all the test cases. Especially for the rectangle object-hole, where it requires an accurate rotation error correction, the success rate was 7% higher, and the attempt number was 24% lower than the previous work. This is because the proposed method takes informative measurement through the controlled motion and efficiently represents it as the explicit contact line estimates. It also has a recurrent structure, which is influenced by estimates of previous steps.

‘w/o Sticking Factor’ shows similar performance as the original since it has a similar contact line estimation performance in the horizontal direction as seen in Section 3.5.1. The performance decreases slightly for ‘w/o Deformation Factor’. ‘w/o Active Motion’ shows poorer performance, especially for the rectangle object-hole case.

3.6 Conclusion and Future Work

We propose a framework for active extrinsic contact sensing and apply it to an insertion policy by using the estimated contact line as an input to an insertion policy trained in simulation with RL. The factor graph-based estimation model and the active tactile feedback controller work collaboratively to localize the contact line between a grasped object and an environment. We then formulate the insertion task as an RL problem where the input is the series of estimated contacts with the hole. This enables us to train the RL agent in simulation and ease the burden of collecting training data with real experiments. In future work, we would like to extend this framework to more general manipulation scenarios: point contacts, non-stationary contacts, and non-flat object/environment surfaces.

Chapter 4

Simultaneous Tactile Estimation and Control:

How Can We Reason About Extrinsic Contact Interaction With Tactile Sensing?

We propose a method that simultaneously estimates and controls extrinsic contact with tactile feedback. The method enables challenging manipulation tasks that require controlling light forces and accurate motions in contact, such as balancing an unknown object on a thin rod standing upright. A factor graph-based framework fuses a sequence of tactile and kinematic measurements to estimate and control the interaction between gripper-object-environment, including the location and wrench at the extrinsic contact between the grasped object and the environment and the grasp wrench transferred from the gripper to the object. The same framework simultaneously plans the gripper motions that make it possible to estimate the state while satisfying regularizing control objectives to prevent slip, such as minimizing the grasp wrench and minimizing frictional force at the extrinsic contact. We show results with sub-millimeter contact localization error and good slip prevention even on slippery environments, for multiple contact formations (point, line, patch contact) and transitions between them. See supplementary video and results at https://sites.google.com/view/sim-tact.

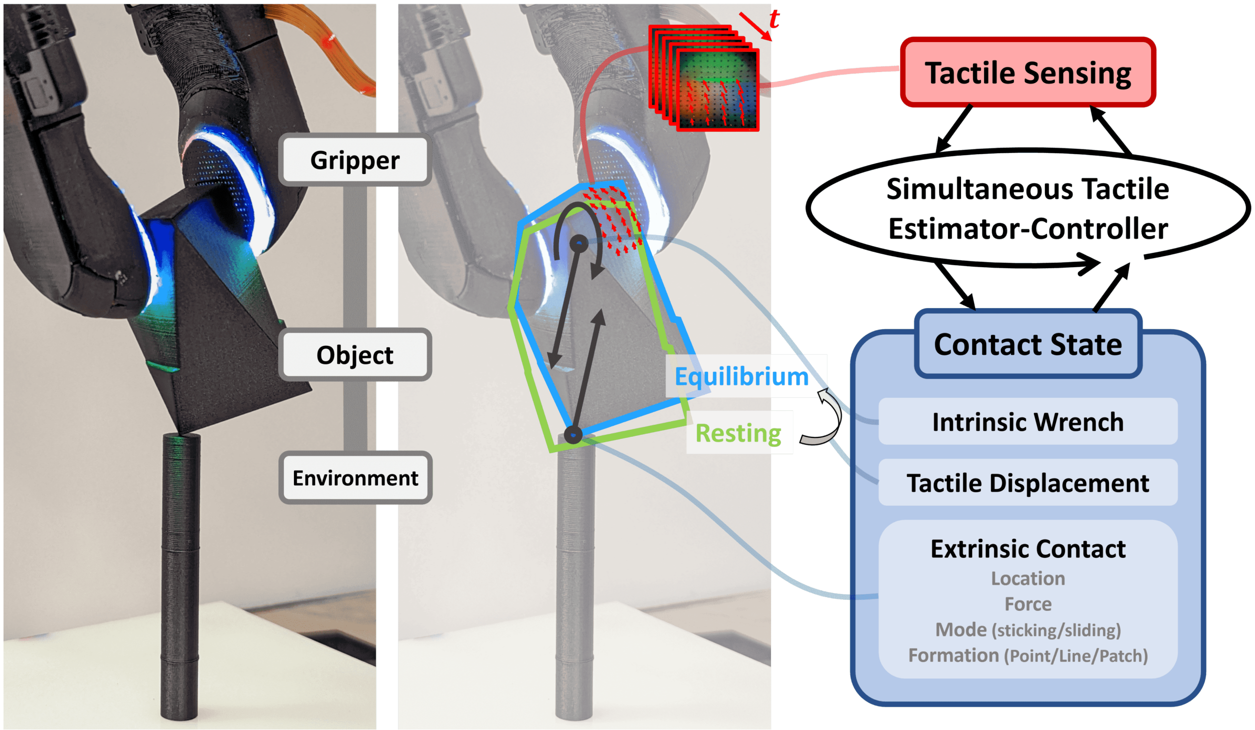


Figure 4.1: Simultaneous tactile estimator-controller used to stably place an unknown object on an unsupported thin rod standing upright.

4.1 Introduction

Tactile feedback is specially useful in manipulation tasks where visual information is limited, or tasks that involve contact with an unknown or uncertain object and environment [42]. We are specially interested in tasks where precise contact regulation is important, but at the same time difficult to observe.

Figure 4.1 shows an illustrative–maybe extreme–example. A gripper is holding an unknown object and is attempting to balance a corner of that object on a free-standing rod. Forces that are transmitted directly vertically at the contact between the object and the rod (*extrinsic contact*) will excite the internal forces of the grasp (*intrinsic contact*) and generate informative signals for estimating the location of that contact. But forces in any other direction are to be avoided, they will make the rod pivot and quickly tumble. When the rod is thin, this is a very difficult–close to impossible–task for a person to control, even with direct line of sight, requiring both compliance and kinematic precision.

In this work we show that, by integrating tactile and kinematic measurements, and with a simultaneous estimation and control framework, it is possible for a robot to do it blind, with arbitrary objects, and with poor prior information of where the external contact is located. Realizing this simultaneous estimation-control behavior with tactile feedback involves the following challenges:

- A single snapshot from tactile sensors does not fully describe the contact configuration or the kinematic state of the system [32]. We need to fuse measurements over time along informative motions to estimate it with some certainty.
- The extrinsic contact between the grasped object and the environment is un-sensed, and to reason about it we need to infer through the chain of contacts that connects the end-effector to the object and to the environment. The tactile sensors we use allow to directly observe the compliance between the object and the end-effector, and tracking that compliance makes it possible to reason about the behavior of the extrinsic contact (as in [25]).

This paper proposes a framework that uses a factor graph-based simultaneous tactile estimator-controller framework to: 1) **estimate** the contact state: object's relative displacement, extrinsic contact location and formation, and the wrench exerted at the intrinsic and extrinsic contacts; and simultaneously, 2) **control** the contact state to the desired configuration. We especially focus on localizing the contact with active motion while maintaining sticking contact between object and environment in multiple contact formations, importantly, doing this while minimizing the forces exerted on the environment.

4.2 Related Work

4.2.1 Contact State Estimation and Control

There has been prior work on contact state estimation and control using tactile sensors [14], [27], [45], [50], [62]. One line of the work is about avoiding slipping and maintaining sticking contact at the intrinsic contact between the sensor finger and the grasped object [52], [63], [64]. While maintaining the sticking contact, other researchers use tactile sensing to control the pose of the manipulated objects when executing manipulation primitives [53], [65]. There also has been research that studies sliding dynamics at the gripper finger [54]. Lastly, researchers have used tactile sensing for planning and decision-making [18], [66]. However, most of these methods focus only on the intrinsic contact state and do not reason carefully about the extrinsic contact state.

Doshi et al. [67] developed a contact configuration estimator and controller to manipulate unknown objects. Their task and approach are similar to ours in that they do both estimation and control. They used force-torque sensing instead of tactile sensing, which eased a challenge mentioned in Sec.4.1. Kim and Rodriguez [68] used tactile sensors to localize the extrinsic contact and solved the peg-in-hole insertion using the estimated contact location. They used a proportional controller to maintain the constant intrinsic finger wrench in order to avoid slippage at extrinsic contact. While both above methods estimate and control the extrinsic contact, they do them separately in independent architectures rather than simultaneously in an integrated architecture.

4.2.2 Factor Graph for Estimation and Control

A factor graph is a bipartite graphical model composed of variables and factors where each factor represents a function on a subset of the variables. One application recently gaining attention is using factor graphs for planning and control [69]. For example, one can construct a linear quadratic regulator (LQR) by representing the state and control input as variables and system dynamics as factors [70]. Dong et al. [71] formulate the Gaussian Process trajectory prior as a chain of variables and factors to do motion planning that results in a smooth trajectory. The same researchers demonstrate simultaneous motion planning (future) and trajectory estimation (past) in the same factor graph [72]. This is especially relevant to our work, where our framework also tries to control (future) and estimate (past) the contact state simultaneously.

4.3 Problem Formulation

The problem we solve is to estimate and control the contact state in multiple contact formations and to detect the transition between the different contact formations. The conditions of the problem are:

- The object is unknown and rigid. The bottom of the object has appropriate shape to make it possible to be placed stably (e.g., corners, edges, flat bottom).
- The environment is flat at the contact with known normal direction.
- At the first contact, the object and the environment meet at a point contact.
- We do not make use of visual feedback.

Specifically, we would like to localize and control the contact between object and environment with minimal slip by minimizing the tangential force at extrinsic contact. To do so, we estimate and control the contact state shown in Figure 4.1:

- gripper-object relative displacement from resting pose to equilibrium pose (tactile displacement)
- location and formation of extrinsic contact

- intrinsic wrench exerted from the gripper to object, and a resultant extrinsic contact force

The state is composed by five quantities:

- $\mathbf{g} \in SE(3)$ - gripper pose
- $\mathbf{o} = \{{}^{r}o, {}^{eq}o\} \in SE(3)$ - object pose
- $\mathbf{w} = \{M_x, M_y, M_z, F_x, F_y, F_z\} \in R^6$ - intrinsic wrench
- $\mathbf{c} \in SE(3)$ - contact pose
- $\mathbf{K} = \{\kappa_x, \kappa_y, \kappa_z, k_x, k_y, k_z, \eta_x, \eta_y, \eta_z\} \in R^9$ - grasp parameters

${}^{r}o$ and ${}^{eq}o$ are the resting and equilibrium object poses. $\mathbf{w}$ is the wrench the gripper exerts on the object. κ and k are the rotational and translational stiffness of the grasp due to the elasticity of the gripper finger. η is the offset of the center of compliance from the gripper center point. The center of compliance is the point where there is no translational displacement when pure torque is exerted on the object, and also it is where we define the intrinsic wrench (w). κ, k, and η are combined as K, which we call “grasp parameters". The grasp parameters vary depending on the local geometry of the object and the grasping force.

For a point contact, we only need translational components of the contact pose, which makes the rotation of $\mathbf{c}$ redundant. To handle this redundancy, we fix the rotation to be aligned with the environment frame. Similarly, we only need one rotational dimension for line contact since we already have that the contact line is on the environment surface. Therefore, we constrain the z-axis of $\mathbf{c}$ to be perpendicular to the environment surface, while the x-axis represents the contact line.

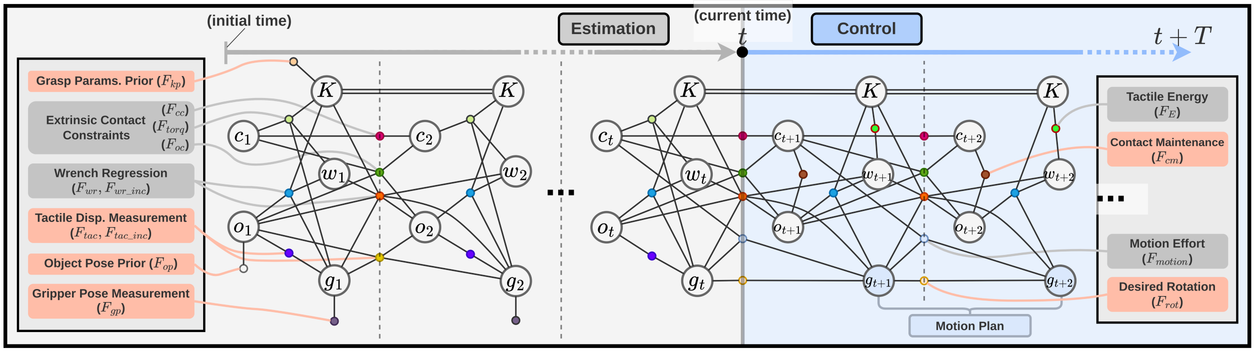

(initial time)
Estimation
(current time)
t
Control
$t+T$
Grasp Params. Prior (F_{kp})
Extrinsic Contact Constraints
(F_{cc})
(F_{torq})
(F_{oc})
Wrench Regression (F_{wr}, F_{wr_inc})
Tactile Disp. Measurement (F_{tac}, F_{tac_inc})
Object Pose Prior (F_{op})
Gripper Pose Measurement (F_{gp})
K
c_1
c_2
w_1
w_2
o_1
o_2
g_1
g_2
...
c_t
w_t
o_t
g_t
c_{t+1}
w_{t+1}
o_{t+1}
g_{t+1}
c_{t+2}
w_{t+2}
o_{t+2}
g_{t+2}
Tactile Energy (F_E)
Contact Maintenance (F_{cm})
Motion Effort (F_{motion})
Desired Rotation (F_{rot})
Motion Plan

Figure 4.2: Factor graph architecture for the simultaneous tactile estimator-controller

4.4 Simultaneous Tactile Estimator-Controller

We use a factor graph to simultaneously estimate and control the contact state with tactile feedback, shown in Figure 4.2. Each circle represents a variable, and each dot represents a factor. Factors with the red labels take input as measurements, prior, or commands. The left part of the architecture, from timestep 1 to t (past), is responsible for aggregating measurements and estimating the past/current contact states. The part from timestep $t+1$ to $t+T$ (future) is accountable for turning the desired rotation of the object into planned gripper motion (blue circles) and predict the future contact states. T is the length of the control horizon. Both parts can be solved simultaneously by minimizing the total factor costs, which breaks down to a nonlinear least-squares problem:

$$\hat{x} = \arg\min_{x} \sum_{f} ||F_f(x_{1:t}, x_{t+1:t+T}; z_{1:t}, z_{t+1:t+T})||^2_{\sum_f} \tag{4.1}$$

where $x_{1:t}$ and $x_{t+1:t+T}$ are past and future states, $z_{1:t}$ is input from measurements and priors, and $z_{t+1:t+T}$ is input from commands. F_f represents factors in Figure 4.2. $\sum_f$ is the covariance of the factors' noise model.

When the new measurement arrives, the factors at timestep $t+1$ are modified, and the new measurement is added, so the border between the estimation and control part is pushed one step forward. Also, one control step is added at $t + T + 1$, so the length of the control horizon remains constant at T. Then, we solve this modified least-squares problem to update the solution. We use an incremental solver [58], which allows fast computation. After each update, the gripper motion plan is sent to the robot, and the robot follows that trajectory until the next update. The update is asynchronous from the robot control; therefore, the robot does not have to wait until the next update unless it reaches the end of the previous motion plan trajectory.

4.4.1 Contact State Estimation

We describe the terms in Eq.5.4 that are responsible for contact state estimation ($\hat{x}_{1:t}$). These terms take inputs from priors and measurements and impose costs for constraints.

Priors: Unary factors impose priors on the variables.

$$F_{kp}(K; K^*) := K - K^* \tag{4.2}$$

$$F_{op}(o_1; o^*) := o^{*-1}({}^r o_1) \tag{4.3}$$

K^* and o^* are the priors for grasp parameters and initial object pose. We obtain grasp parameters prior by fitting it to one randomly selected grasp on the rectangular test object. Then, the same unmodified prior is used for other grasps/objects. A weak initial object pose prior is selected to have a bottom surface parallel to the environment because we do not know object shape and orientation.

Gripper Pose Measurement: The gripper pose measurement from forward kinematics (g_i^*) is imposed as a unary factor:

$$F_{gp}(g_i; g_i^*) := g_i^{*-1} g_i \tag{4.4}$$

Tactile Displacement Measurement: To measure the displacement of the object from resting to equilibrium pose under contact, we use GelSlim 3.0, a vision-based tactile sensor that observes the deformation of the sensor finger due to contact as a high-resolution tactile image. We feed the tactile image to the tactile module adopted from [68] to get the object displacement (δ_i). The tactile module is a convolutional neural network that takes the tactile image as input and outputs the tactile displacement. We use two types of factors to impose the measurement: one for total displacement (F_{tac}) and the other for incremental displacement (F_{tac_inc}):

$$F_{tac}(g_i, o_i; \delta_i) = \delta_i^{-1}(({}^{r}o_i^{-1} g_i)^{-1}({}^{eq}o_i^{-1} g_i)) \tag{4.5}$$

$$\begin{aligned} &F_{tac\_inc}(g_{i-1}, o_{i-1}, g_i, o_i; \delta_{i-1}, \delta_i) \\ &\qquad = (\delta_{i-1}^{-1}\delta_i)^{-1}(({}^{eq}o_{i-1}^{-1} g_{i-1})^{-1}({}^{eq}o_i^{-1} g_i)) \end{aligned} \tag{4.6}$$

Extrinsic Contact Geometric Constraints:

$$F_{oc}(o_{i-1}, c_{i-1} o_i, c_i) = ({}^{eq}o_{i-1}^{-1} c_{i-1})^{-1}({}^{eq}o_i^{-1} c_i) \tag{4.7}$$

$$F_{cc}(c_{i-1}, c_i) = c_{i-1}^{-1} c_i \tag{4.8}$$

$\mathbf{F_{oc}}$: Unless there is a transition in the contact formation, the location of contact in the object frame should remain constant. We impose it as a strong factor between adjacent timesteps. $\mathbf{F_{cc}}$: Since we assume a flat environment, the location of contact on the environment should not change in the direction perpendicular to the environment surface. Also, the change in the tangential direction should be small if the control objective of minimizing slip is met properly. We impose this by formulating a binary factor and setting a strong cost in the perpendicular direction and less strong in the tangential direction.

Extrinsic Contact Torque Constraints: During point contact, there should not be torque exerted about the contact, and during line contact, there should not be torque component in the direction parallel to the contact line.

$$\begin{aligned} {}^{p}F_{torq}(g_i, w_i, c_i, K) &= \vec{M} - \vec{r}(g_i, c_i, \eta) \times \vec{F} \\ &= \vec{M} - ((g_i^{-1} c_i)_{trn} - \eta) \times \vec{F} \quad \text{(point contact)} \\ {}^{l}F_{torq}(g_i, w_i, c_i, K) &= {}^{p}F_{torq} \cdot \vec{a}_x(g_i, c_i) \quad \text{(line contact)} \end{aligned} \tag{4.9}$$

where $(g_i^{-1} c_i)_{trn}$ is the translational location of the contact with respect to the gripper, and $\vec{a}_x(g_i, c_i)$ is the unit vector parallel to the estimated contact line.

Wrench Regression Constraints: To estimate the intrinsic wrench from the measurements, we approximate the grasp as decoupled linear springs in each rotational and translational direction, with an additional nonlinear term Δ:

$$w_i = [\ \vec{M},\ \vec{F}\ ] = [\ \kappa,\ \mathbf{k}\ ] \odot ({}^{r}o_i^{-1} g_i)^{-1} ({}^{eq}o_i^{-1} g_i) + \Delta \tag{4.10}$$

where $\odot$ is element-wise multiplication. $({}^{r}o_i^{-1} g_i)^{-1}({}^{eq}o_i^{-1} g_i)$ is the tactile displacement in a canonical coordinate $[R_x, R_y, R_z, x, y, z]$, which we dropped the coordinate notation for simplicity. We regress the wrench close to the linear relation by imposing two types of factors:

$$F_{wr}(g_i, o_i, w_i, K) = w_i - [\kappa, \mathbf{k}] \odot ({}^{r}o_i^{-1} g_i)^{-1} ({}^{eq}o_i^{-1} g_i) \tag{4.11}$$

$$\begin{aligned} &F_{wr\_inc}(g_{i-1}, o_{i-1}, w_{i-1}, g_i, o_i, w_i, K) \\ &\quad = (w_i - w_{i-1}) - [\ \kappa,\ \mathbf{k}\ ] \odot ({}^{eq}o_{i-1}^{-1} g_{i-1})^{-1} ({}^{eq}o_i^{-1} g_i) \end{aligned} \tag{4.12}$$

F_{wr} penalize the additional nonlinear term. F_{wr_inc} tries to regress the incremental change in the intrinsic wrench to be parallel to the linear relation.

4.4.2 Contact State Control

We describe the terms in Eq.5.4 that are responsible for contact state control and prediction ($\hat{x}_{t+1:t+T}$). In a nutshell, the system achieves the behavior of pivoting about an unknown contact point or line as the combination of these objectives: 1) A desired rotation of the object; 2) Minimizing motion effort and tactile deformation; 3) Maintaining a minimum contact with the environment.

Desired Rotation: While the factor graph computes the fine motion, we still need a input command in which direction we want to rotate (tilt) the object:

$$F_{rot}(g_{i-1}, g_i; R_i) = R_i^{-1}(g_{i-1}^{-1}g_i)_{rot} \tag{4.13}$$

where R_i is the desired rotation. This factor is tuned to be weaker than other control objectives, so the actual rotation can deviate from the desired rotation if this factor conflicts with other control objective factors (e.g., tactile energy). For example, in Figure 4.4d, the actual rotation (solid line) deviates from the desired rotation (dashed line).

Motion Effort: We impose a cost on the local motion at the estimated contact point:

$$F_{motion}(g_{i-1}, g_i, c_{i-1}) = c_{i-1}^{-1}g_i g_{i-1}^{-1}c_{i-1} \tag{4.14}$$

Tactile Energy: As we regress the intrinsic wrench to the decoupled linear relation, we approximate the elastic potential energy on the sensor finger as the quadratic sum of wrench components and impose it as a factor:

$$
\begin{aligned}
E &= \frac{1}{2}\left\{\frac{M_x^2}{\kappa_x} + \frac{M_y^2}{\kappa_y} + \frac{M_z^2}{\kappa_x} + \frac{F_x^2}{k_x} + \frac{F_y^2}{k_y} + \frac{F_z^2}{k_z}\right\} \qquad (4.15)\\
&\Rightarrow F_E(w_i, K) = w_i \oslash [\ \sqrt{\kappa},\ \sqrt{\mathbf{k}}\ ] \qquad (4.16)
\end{aligned}
$$

where $\oslash$ is element-wise division.

This factor plays two important roles. First, it enables the prediction of future tactile displacement. This is because, given the extrinsic contact as a constraint, the intrinsic contact will be at a stable state where it minimizes the potential energy. Also, this factor helps to achieve control objectives while having minimal deformation on the sensor finger.

Contact Maintenance: None of the above control objective factors force the robot to push the object against the environment. In fact, minimizing tactile energy encourages moving away from the environment. Therefore, we impose a factor that encourages the robot to push the object to the environment by setting the target contact an offset distance inside the environment (ϵ_i). We use a hinge function that penalizes when the estimated offset distance is less than the desired distance:

$$
\begin{aligned}
F_{cm}(o_i, c_i; \epsilon_i) &= \max(0, \zeta_i(o_i, c_i) - \epsilon_i)\\
&= \max(0, -(({}^{r}o_i^{-1}c)^{-1}({}^{eq}o_i^{-1}c))_{trn,z} - \epsilon_i) \qquad (4.17)
\end{aligned}
$$

where ζ_i is the estimated offset distance.

When this factor is combined with the tactile energy factor, it enables minimizing the tangential force at the extrinsic contact. Imagine the contact state with no tangential force at the extrinsic contact. If we move the gripper parallel to the environment surface by applying tangential force in any direction, it will do positive work, which adds to the tactile energy while maintaining the constant offset distance. In other words, if the offset distance

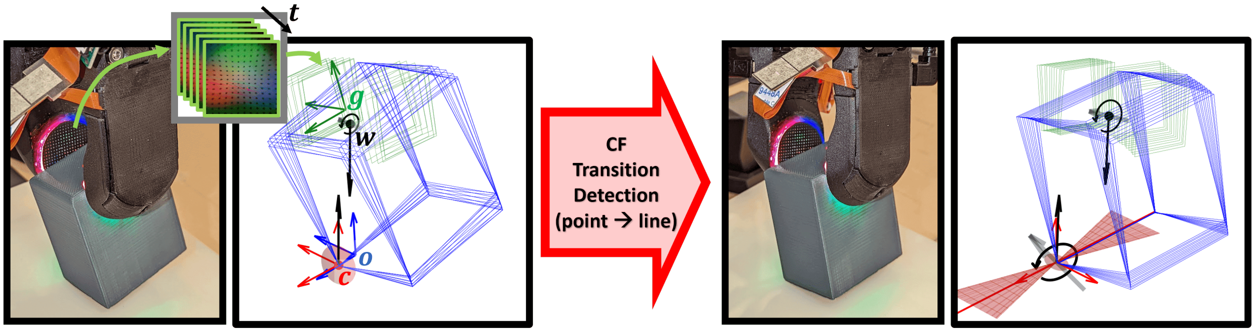


Figure 4.3: An object making point (left) and line (right) contact with the horizontal environment

is fixed, minimizing the tactile energy is equivalent to minimizing the tangential force.

4.4.3 Detection of Contact Formation Transition

When the contact formation transitions from one to another, the **extrinsic contact torque constraints** are violated. Therefore, the transition can be detected by simply measuring an increase in the residual in Eq.4.9. Figure 4.4e shows an example of detecting transitions from point-to-line and line-to-patch, where the horizontal line at 0.1 is the threshold for the detection. After the transition, we can continue to use the estimator-controller by modifying the noise model of the factors accordingly.

4.5 Experiments and Results

Figure 4.5 shows the hardware setup for experiments. We use 6-DoF ABB's IRB120 robot arm and a WSG-50 gripper mounted with GelSlim 3.0 [27] on each side of the finger. ATI's Net Force-Torque sensor is mounted beneath the environment to collect the extrinsic contact force. We use eight 3D printed test objects, including rectangular and hexagonal cylinders and six others with irregular shapes (Figure 4.7), to evaluate the generalization to various grasps and geometries. Pytorch [73] is used for the tactile module computation, and iSAM2 [58] with GTSAM library [59] is used to solve the least-squares problem incrementally.

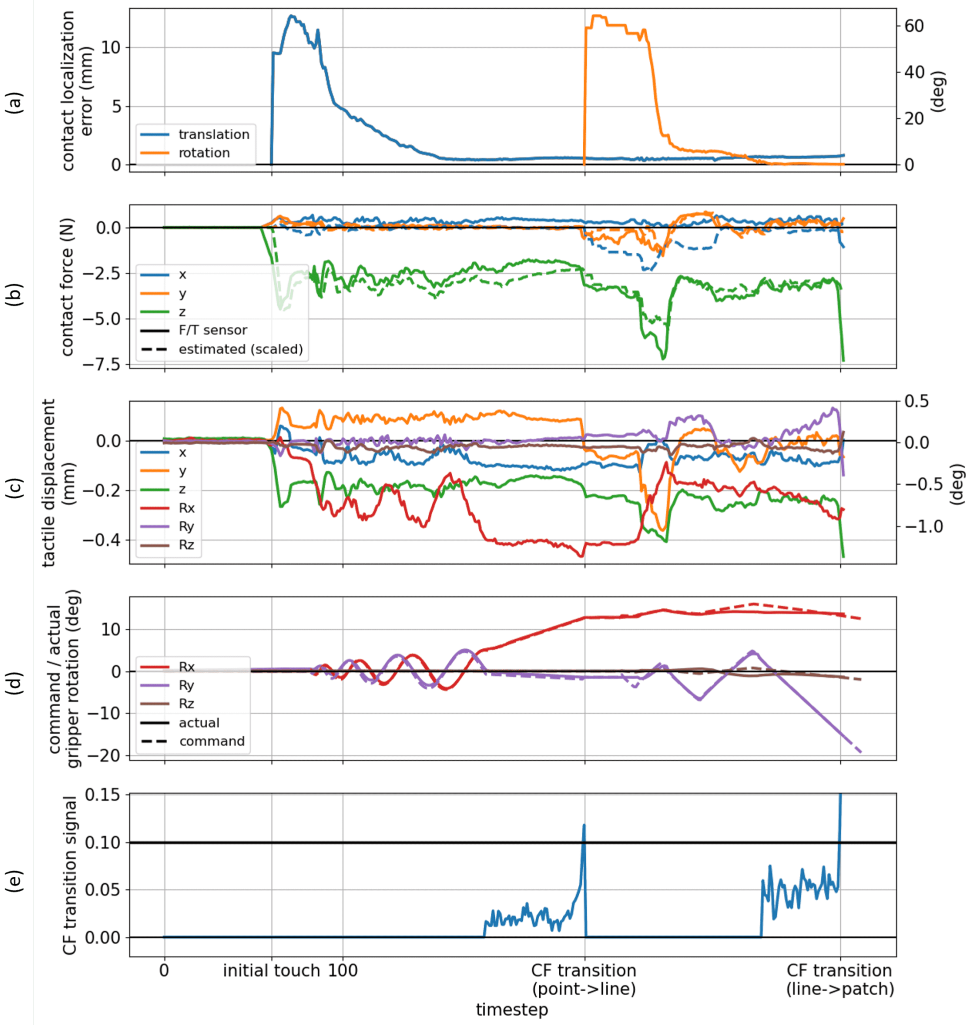


Figure 4.4: Time series plots for the case in Figure 4.3

4.5.1 Point Contact Formation

We test our method in a point contact formation to evaluate localization accuracy and capability to minimize the tangential contact force and slippage. We first grasp and orient the test objects with a random pose. Then we start a desired rotation command that draws a **small** conical spiral-like trajectory with a maximum of 5 degrees angle deviation from the initial orientation. Then we continue with a **large** cone-like desired rotation command with a maximum of 15 degrees from the initial orientation. We change the environment surface material between a higher friction paper, where we can evaluate the tangential force, and

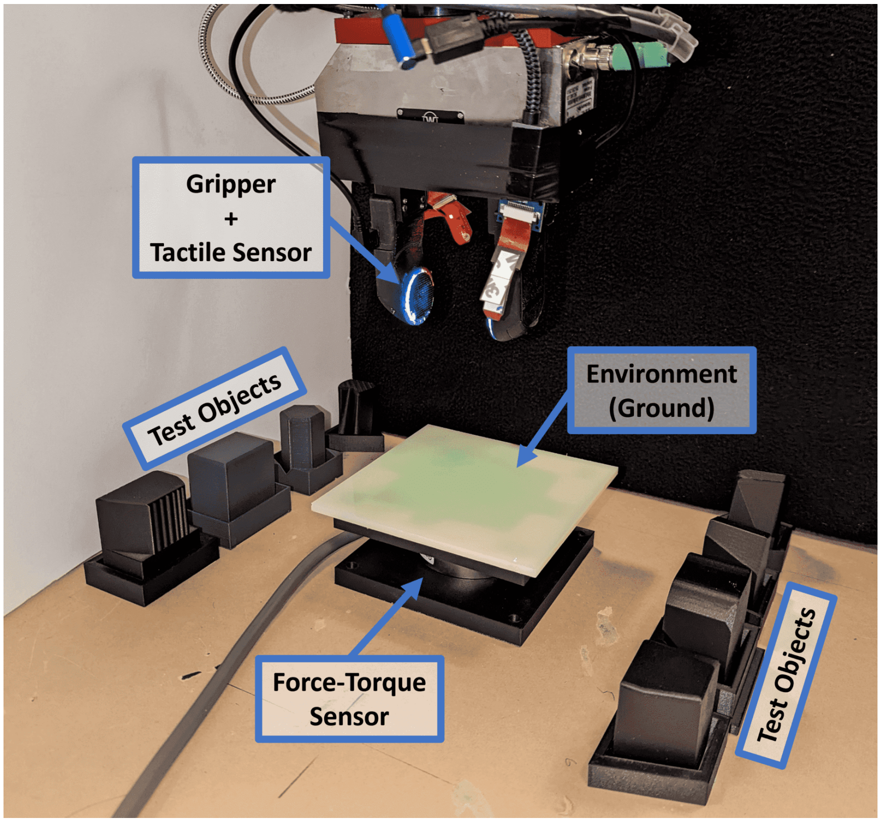


Figure 4.5: Experimental setup

a lower friction acrylic, where we can evaluate the slippage. The result with 20 trials per each object is shown in Table.4.1 and Figure 4.6. We show localization error and mean/max tangential to normal force ratio during the motion for the higher friction surface. For the lower friction surface, we show localization error with total slipped distance from the initial contact location. We report the result from two ablation models: the 'constant tactile' and the 'w/o tactile E.' The 'constant tactile' is the model in [68]. It uses a proportional controller to keep the tactile displacement constant throughout the motion. In other words, it tries to maintain a constant intrinsic wrench. The 'w/o tactile E' is similar to our proposed method, but we make the weight of the tactile energy factor close to zero, so it will not prioritize minimizing the tactile energy.

On the higher friction surface, the proposed method showed approximately 25% and 40% reduction in localization error than the 'constant tactile' and 'w/o tactile E,' respec-

Table 4.1: Evaluation of Simultaneous Tactile Estimator-Controller on Point Contact Formation

(a) Higher Friction (Paper)

	Rectangle					Hexagon				
	loc. error (mm)	tan./norm. (mean)		tan./norm. (max)		loc. error (mm)	tan./norm. (mean)		tan./norm. (max)	
		small motion	large motion	small motion	large motion		small motion	large motion	small motion	large motion
proposed	**0.89**	0.105	0.096	0.201	**0.193**	**0.81**	0.060	0.091	0.134	**0.188**
constant tactile	**0.94**	0.172	0.176	0.252	**0.278**	**1.30**	0.056	0.132	0.102	**0.282**
w/o tactile E	**1.42**	0.180	0.216	0.249	**0.330**	**1.38**	0.047	0.156	0.103	**0.303**

(b) Lower Friction (Acrylic)

	Rectangle			Hexagon		
	loc. error (mm)	slipped distance (mm)		loc. error (mm)	slipped distance (mm)	
		small motion	large motion		small motion	large motion
proposed	**0.90**	0.19	**0.35**	**0.78**	0.06	**0.16**
constant tactile	**1.93**	0.84	**12.90**	**1.43**	0.06	**4.27**

tively. The difference is more significant in tangential to normal force ratio that the error bar of the proposed method does not overlap with those of ablation methods (Figure 4.6) Hence, the proposed method is better at reducing the tangential forces required for pivoting, and less likely to slip.

On the lower friction surface, the proposed method is compared with the 'constant tactile' to evaluate the slip minimization capability. The 'constant tactile' showed small slippage with small motion but showed about two orders of magnitude higher slippage with large motion. This is because maintaining the constant intrinsic wrench will not be sufficient to keep the extrinsic contact force inside the friction cone if the motion is large. As a result of the large slip, the accuracy of localization also decreases. The 'constant tactile' shows about two times larger localization error than the proposed method.

Contact Force Estimation: From the estimated intrinsic wrench (w), we estimate the

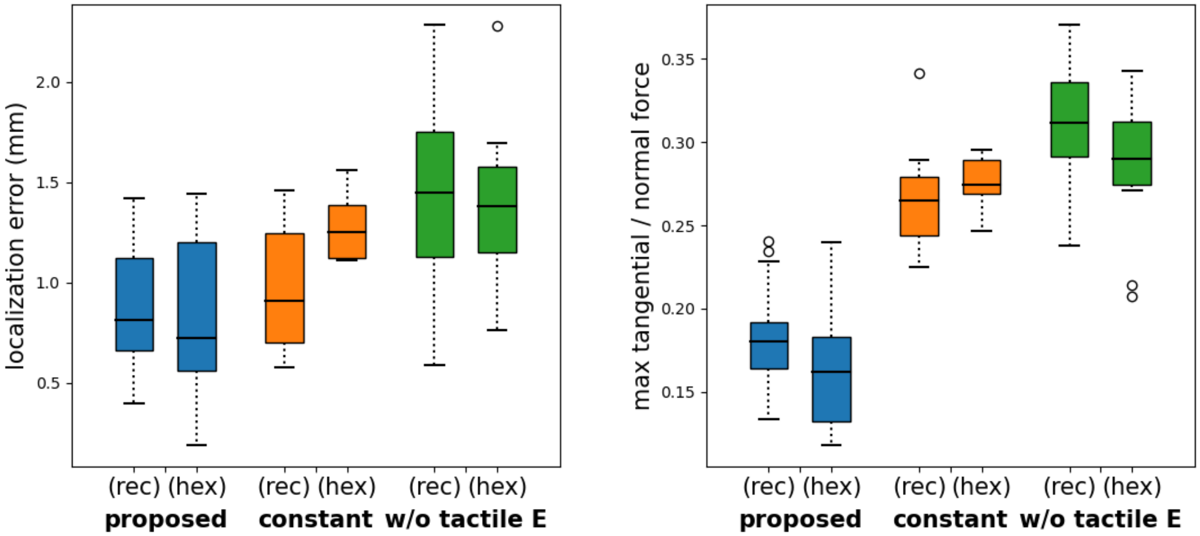


Figure 4.6: (left) Localization error, (right) maximum tangential to normal force ratio during the point contact motion on the higher friction surface

extrinsic contact force (e.g., dashed line in Figure 4.4b) and evaluate the accuracy of the force estimation by comparing it with the F/T sensor measurement on the environment. However, the estimated force does not have a physical unit specified, so we only compare the mean difference in angle between the estimated force and the sensor measurement over all the data collected. To study the effect of incorporating the nonlinear term (Δ) in Eq.4.10, we evaluate the accuracy without adding the nonlinear term. To study the effect of estimating the grasp parameters (K) rather than using initial prior, we use the initial prior values instead of estimated values and also remove the nonlinear term in Eq.4.10, then evaluate the accuracy. The result is shown in Table.4.2. The performance gets better as we use the estimated grasp parameters instead of the initial prior values and also as the nonlinear term is considered. This implies that considering the nonlinear term and estimating the grasp parameters effectively improves the force estimation.

4.5.2 Multiple Contact Formations

We test the proposed method on multiple contact formations, which requires the ability to detect the transition between different contact formations. Working in multiple contact

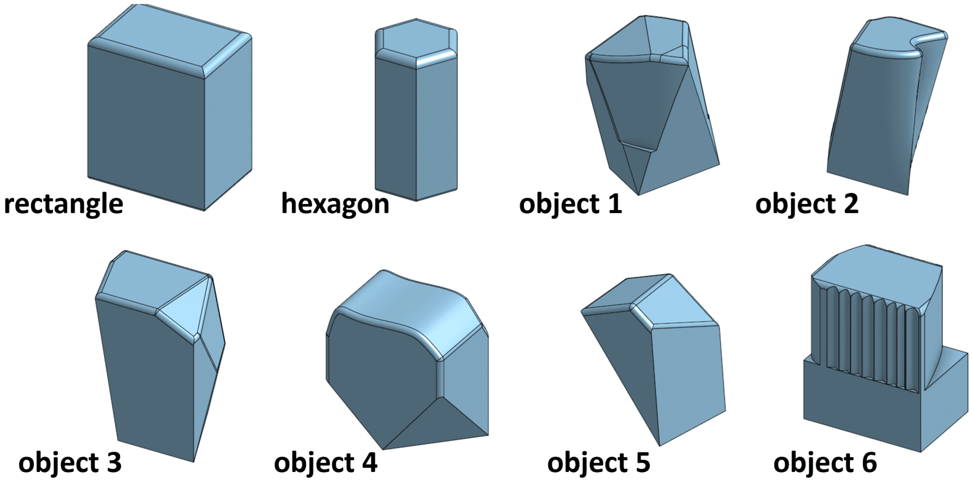


Figure 4.7: CAD model of the test objects

Table 4.2: Contact Force Estimation Accuracy

	Mean Misalignment (deg)	
	Rectangle	Hexagon
proposed	6.2	5.0
w/o nonlinear term	9.3	13.0
w/o nonlinear term & grasp parameters estimation	13.9	14.5

formations is important in many manipulation tasks. Figure 4.3 and Figure 4.4 show an example of placing the rectangular object on a flat surface. It first makes initial point contact with the surface, executes motion to localize the point contact, detects the transition to the line contact, localizes the line contact, then finally detects the transition to the patch contact before releasing the grasp.

We run a similar experiments on the lower friction surface with all eight 3D printed objects. We focus on point and line contact because patch contact transition is trivial as it does not require or allow any motion after the transition. We first execute the same small desired rotation command as in Sec.4.5.1, then command the desired rotation to tilt roughly towards the direction where the object edge is. We assume we have a weak prior knowledge of the object edge's direction by adding a maximum error of 20 degrees to the true direction.

Table 4.3: Evaluation of Simultaneous Tactile Estimator-Controller on Multiple Contact Formation

	Point Contact		Successful Transition Rate	Line Contact		
	loc. err. (mm)	slip (mm)		loc. err. (deg)	slip	
					trans. (mm)	rot. (deg)
rectangle	0.80	0.77	10 / 10	0.81	1.31	1.85
hexagon	0.76	0.48	8 / 10	4.54	0.84	1.16
object 1	0.99	0.63	8 / 10	2.66	1.73	1.75
object 2	1.19	0.87	6 / 10	4.24	3.41	1.98
object 3	0.96	0.94	10 / 10	1.58	1.18	1.48
object 4	0.93	1.08	9 / 10	0.82	1.12	1.39
object 5	0.86	0.47	8 / 10	2.03	1.25	1.70
object 6	0.93	0.75	10 / 10	1.84	0.83	1.00

After detecting the point-to-line contact transition, we modify the desired command rotation to draw a sinusoidal-like shape, where the sinusoid amplitude is perpendicular to the direction of tilt before the transition (dashed line in Figure 4.4d). We estimate the line localization accuracy when the amount of rotation from the pose at the transition reaches 5 degrees.

Table.4.3 shows the localization error, amount of slippage, and successful transition rate of the point-to-line contact transition. The failed transition includes the case where the transition is detected too early or late, so it loses the line contact, which leads to poor line estimation accuracy or causes the least-squares solver to throw an indeterminant error. The line contact evaluation metrics are calculated with only the successful cases.

Most test objects showed reasonably good localization errors, a small amount of slippage, and a high successful transition rate, while hexagon and object 2 showed relatively higher line localization errors. Object 2 also showed a lower successful transition rate and higher slippage amount than others. One main reason the hexagonal object showed higher line localization error is that it has a shorter edge than other objects. The hexagon edge is 17.5

mm, which is significantly shorter than other object edges; for example, the shortest edge of the rectangle is 35 mm. A short edge will make it harder for our method to detect the point-to-line contact transition, and it is more likely that the object loses the line contact with the surface during motion. Object 2's worse performance is likely due to the different local grasp geometry. It has a high curvature and sharp cut at the grasped part, which leads to a smaller contact patch with the fingers. This might makes the grasp parameters (K) deviate much from that of other objects and lead to worse performance.

4.6 Conclusion and Future Work

We demonstrate simultaneous estimation and control for contact state of unknown objects using tactile sensing. We show that we can localize the extrinsic contact by fusing the gripper pose and tactile measurements in a factor graph. Regressing the grasp mechanics to a decoupled stiffness model of the grasp enables the estimation of the grasp wrench and the contact force. Also, we impose control objectives as factors in the same graph to execute the pivoting motion with minimal tangential force and slip. Lastly, we implement the method on multiple contact formations and are able to successfully detect the transition between them.

While the proposed work focuses only on minimizing the tangential forces on the environment surface, one possible follow-up is reasoning about normal force. A trade-off is that a larger normal force will enhance observability but possibly stress the environment and object. Extending the proposed work to not only the sticking contact but other contact modes is also an important direction of research. Lastly, we plan to develop an automated desired command rotation policy instead of manually commanding it.

Chapter 5

TEXterity - Tactile Extrinsic deXterity:

How Can We Achieve Extrinsic Dexterity With Tactile Sensing?

We introduce a novel approach that combines tactile estimation and control for in-hand object manipulation. By integrating measurements from robot kinematics and an image-based tactile sensor, our framework estimates and tracks object pose while simultaneously generating motion plans to control the pose of a grasped object. This approach consists of a discrete pose estimator that tracks the most likely sequence of object poses in a coarsely discretized grid, and a continuous pose estimator-controller to refine the pose estimate and accurately manipulate the pose of the grasped object. Our method is tested on diverse objects and configurations, achieving desired manipulation objectives and outperforming single-shot methods in estimation accuracy. The proposed approach holds potential for tasks requiring precise manipulation and limited intrinsic in-hand dexterity under visual occlusion, laying the foundation for closed-loop behavior in applications such as regrasping, insertion, and tool use. Please see https://sites.google.com/view/texterity for videos of real-world demonstrations.

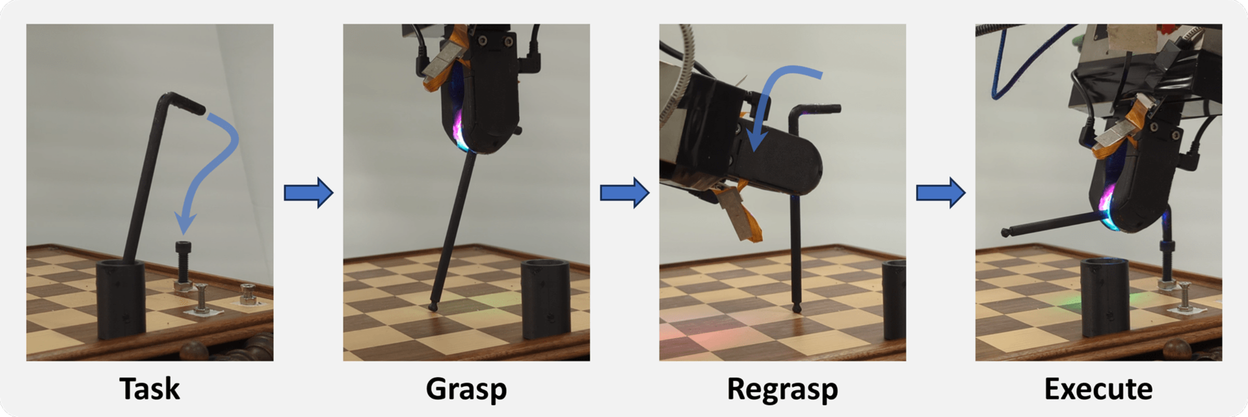


Figure 5.1: An example task that requires tactile extrinsic dexterity. A proper grasp is essential when using an Allen key to apply sufficient torque while fastening a hex bolt. The proposed method utilizes tactile sensing on the robot's finger to localize and track the grasped object's pose and also regrasp the object in hand by pushing it against the floor - effectively leveraging extrinsic dexterity.

5.1 Introduction

The ability to manipulate objects within the hand is a long-standing objective in robotics for its potential to increase the workspace, speed, and capability of robotic systems. For example, the ability to change the grasp on an object can improve grasp stability and functionality, or prevent collisions and kinematic singularities. In-hand manipulation is challenging from the perspectives of state estimation, planning, and control: first, once the object is enveloped by the grasp, it becomes difficult to perceive with external vision systems; second, the hybrid dynamics of contact-rich tasks are difficult to predict [74] and optimize over [75].

Existing work on in-hand manipulation emphasizes the problem of sequencing contact modes, and can be broken down into two prevailing methodologies. One line of work relies on simple object geometries and exact models of contact dynamics to plan using traditional optimization-based approaches [75]–[79], while the other leverages model-free reinforcement learning to learn policies directly that only consider or exploit contact modes implicitly [80]–[85]. Much less consideration has been given to the challenge of precisely controlling such behaviors, despite the fact that prominent tasks like connector insertion or screwing in a small bolt require high precision.

Tactile feedback is a promising modality to enable precise control of in-hand manipulation. Image-based tactile sensing [14], [27], [50] has gained traction in recent years for its ability to provide high-resolution information directly at the contact interface. Image-based tactile sensors have been used for pose estimation [19], object retrieval [86], and texture recognition [87]. They have also been used to estimate the location of contacts with the environment [15], [25], [88], to supervise insertion [18], and to guide the manipulation of objects like boxes [53], tools [89], cable [90], and cloth [91].

We study the problem of precisely controlling in-hand sliding regrasps by pushing against an external surface, i.e. extrinsic dexterity [92], supervised only by robot proprioception and tactile sensing. Our framework is compatible with arbitrary, but known, object geometries and succeeds even when the contact parameters are known only approximately.

This work builds upon previous research efforts. First, *Tac2Pose* [19] estimates the relative gripper/object pose using tactile sensing, but lacks control capabilities. Second, *Simultaneous Tactile Estimation and Control of Extrinsic Contact* [16] estimates and controls extrinsic contact states between the object and its environment, but has no understanding of the object's pose and therefore has limited ability to reason over global re-configuration. Our approach combines the strengths of these two frameworks into a single system. As a result, our method estimates the object's pose and its associated contact configurations and simultaneously controls them. By merging these methodologies, we aim to provide a holistic solution for precisely controlling general planar in-hand manipulation.

This paper is an extension of our work on *tactile extrinsic dexterity* [17] in these ways:

- In Section 5.4.2, we evaluate our method against five ablations for four distinct types of goal configurations. These new results illuminate key features of our approach. In particular, we evaluate the effectiveness of leveraging prior knowledge of the external environment to collapse ambiguity in individual tactile images. In addition, we compare our results against those derived from idealized simulations and using privileged information, to showcase the capability of our approach in bridging the sim-to-real gap.

- In Section 5.4.3, we provide qualitative results for three household objects in realistic scenarios. These results motivate the work concretely, and demonstrate that our method generalizes to real objects, which have a variety of material, inertial, and frictional properties.
- Finally, we provide a more complete review of prior work in Section 5.2, and more thorough explanation of our method in Sections 5.3.3 and 5.3.4.

5.2 Related Work

5.2.1 Tactile Estimation and Control.

Image-based tactile sensors are particularly useful for high-accuracy pose estimation, because they provide high-resolution information about the object geometry throughout manipulation. They have been successfully used to track object drift from a known initial pose [49], [93], build a tactile map and localize the object within it [46], [94], [95], and estimate the pose of small parts from a single tactile image [96]. Because touch provides only local information about the object geometry, most tactile images are inherently ambiguous [19]. Some work has combined touch with vision [97]–[100] to resolve such ambiguity. Our approach is most similar to a line of work that estimates distributions over possible object pose from a single tactile image [19], [101], [102], then fuses information over streams of tactile images using particle [101] or histogram [102] filters. [101] tackles the estimation, but not control, problem, assuming that the object is rigidly fixed in place while a human operator slides a tactile sensor along the object surface. Similarly, [102] also assumes the object is fixed in place, while the robot plans and executes a series of grasp and release maneuvers to localize the object. Our work, on the other hand, tackles the more challenging problem of estimating and controlling the pose of an object sliding within the grasp while not rigidly attached to a fixture. The mechanics of sliding on a deformable sensor surface are difficult to predict, which places more stringent requirements on the quality of the observation model and controller.

5.2.2 In-Hand Manipulation.

In-hand manipulation is most commonly achieved with dexterous hands or by leveraging the surrounding environment (extrinsic dexterity [92]). One line of prior work formulates the problem as an optimization over exact models of the hand/object dynamics [75]–[79], [103], but only for simple objects and generally only in simulation [75], [76], or by relying on accurate knowledge of physical parameters to execute plans precisely in open loop [77]. Another line of prior work focuses on modeling the mechanics of contact itself in a way that is useful for planning and control, either analytically [104]–[106] or with neural networks [107], [108].

Some work has avoided the challenges of modeling contact altogether, instead relying on model-free reinforcement learning with vision to directly learn a policy for arbitrary geometries. Some policies have been tested on simulated vision data only [80], [81], while others operate on real images [82]–[85]. They, however, suffer from a lack of precision. As an example, [82] reports 45% success on held out objects, and 81% success on training objects, where success is defined as a reorientation attempt with less than 0.4 rad (22.9) of error, underscoring the challenge of precise reorientation for arbitrary objects.

There have also been a number of works leveraging tactile sensing for in-hand manipulation. [90], [91], and [65] use image-based tactile sensors to supervise sliding on cables, cloth, and marbles, respectively. [89] detects and corrects for undesired slip during tool manipulation, while [109] learns a policy that trades off between tactile exploration and execution to succeed at insertion tasks. Some works rely on proprioception [110] or pressure sensors [111] to coarsely reorient objects within the hand. State estimation from such sensors is challenging and imprecise, leading to policies that accrue large errors. Another line of work uses tactile sensing to reorient objects within the hand continuously [112]–[116], without considering the challenge of stopping at goal poses precisely.

We consider the complementary problem of planning and controlling over a known contact mode (in-hand sliding by pushing the object against an external surface), where the object geometry is arbitrary but known. We leverage a simple model of the mechanics of sliding and supervise the behavior with high-resolution tactile sensing, in order to achieve precise in-hand manipulation. By emphasizing the simultaneous estimation and control for a realistic in-hand manipulation scenario, this work addresses a gap in the existing literature and paves the way for executing precise dexterous manipulation on real systems.

5.2.3 Extrinsic Contact Estimation and Control.

Extrinsic contacts, or contacts between a grasped object and the surrounding environment, are fundamental to a range of contact-rich tasks including insertion, tool use, and in-hand manipulation via extrinsic dexterity. A variety of work has explored the ability to estimate [25], [88], [117] and control [15], [16], [67], [118] such contacts using intrinsic (on the robot) sensing.

[67] manipulates unknown objects by estimating and controlling extrinsic contacts with force-torque feedback. [25] uses image-based tactile feedback with a small exploratory motion to localize an extrinsic point contact that is fixed on the environment. [15], [16] estimates and controls extrinsic contacts represented as points, lines, and patches with feedback from image-based tactile sensors.

Another line of prior work instead represents and estimates extrinsic contacts using neural implicit functions with tactile [88], [118] or visuo-tactile [117] sensing. Finally, [119] estimates extrinsic contacts from a scene-level RGB-D images of the robot workspace. These methods are complementary to our approach, which explicitly represents the extrinsic contacts using a kinematic model, rather than using implicit neural representations of the extrinsic contacts.

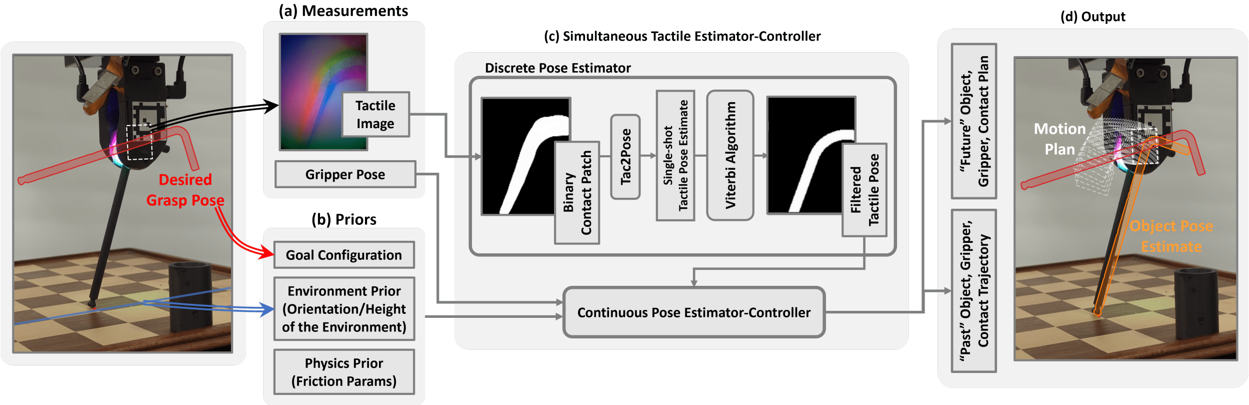

(a) Measurements
Tactile Image
Gripper Pose
(b) Priors
Goal Configuration
Environment Prior (Orientation/Height of the Environment)
Physics Prior (Friction Params)
Desired Grasp Pose
(c) Simultaneous Tactile Estimator-Controller
Discrete Pose Estimator
Binary Contact Patch
Tac2Pose
Single-shot Tactile Pose Estimate
Viterbi Algorithm
Filtered Tactile Pose
Continuous Pose Estimator-Controller
(d) Output
"Future" Object, Gripper, Contact Plan
"Past" Object, Gripper, Contact Trajectory
Motion Plan
Object Pose Estimate

Figure 5.2: Overview of the Simultaneous Tactile Estimation and Control Framework.

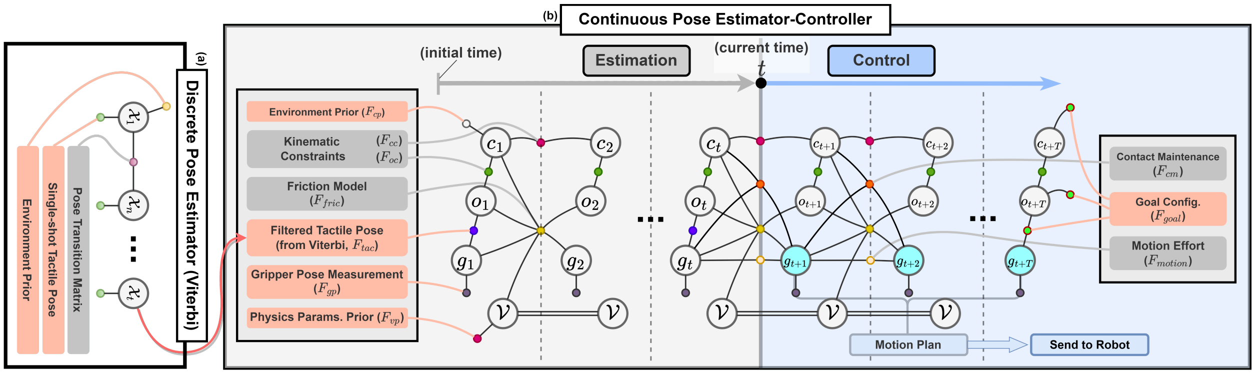

(a) Discrete Pose Estimator (Viterbi)
Environment Prior
Single-shot Tactile Pose
Pose Transition Matrix
x_1
x_n
x_t
(b) Continuous Pose Estimator-Controller
(initial time)
Estimation
(current time)
t
Control
Environment Prior (F_{ep})
Kinematic Constraints (F_{cc}) (F_{oc})
Friction Model (F_{fric})
Filtered Tactile Pose (from Viterbi, F_{tac})
Gripper Pose Measurement (F_{gp})
Physics Params. Prior (F_{vp})
c_1 c_2 o_1 o_2 g_1 g_2 $\mathcal{V}$
c_t c_{t+1} c_{t+2} c_{t+T}
o_t o_{t+1} o_{t+2} o_{t+T}
g_t g_{t+1} g_{t+2} g_{t+T}
Contact Maintenance (F_{cm})
Goal Config. (F_{goal})
Motion Effort (F_{motion})
Motion Plan
Send to Robot

Figure 5.3: Graph Architecture of the Simultaneous Tactile Estimator-Controller.

5.3 Method

5.3.1 Problem Formulation

We address the task of manipulating objects in-hand from unknown initial grasps to achieve desired configurations by pushing against the environment. The target configurations encompass a range of potentially simultaneous manipulation objectives:

- Changing the grasp pose (i.e., relative rotation/translation between the gripper and the object)
- Changing the orientation of the object in the world frame (i.e., pivoting against the environment)

- Changing the location of the extrinsic contact point (i.e., sliding against the environment)

A wide variety of regrasping tasks can be specified via a combination of the above objectives.

We make several assumptions to model this problem:

- Grasped objects are rigid with known 3D models.
- The part of the environment that the object interacts with is flat, with a known orientation and height.
- Contact between the grasped objects and the environment occurs at a single point.
- Grasp reorientation is constrained to the plane of the gripper finger surface.

5.3.2 Overview

Figure 5.1 illustrates our approach through an example task: using an Allen key to apply sufficient torque while fastening a hex bolt. Adjusting the grasp through in-hand manipulation is necessary to increase the torque arm and prevent the robot from hitting its motion limit during the screwing.

Figure 5.2 provides an overview of the framework of our approach. The system gathers measurements from both the robot and the sensor (Figure 5.2a). Robot proprioception provides the gripper's pose, while the GelSlim 3.0 sensor [27] provides observation of the contact interface between the gripper finger and the object in the form of an RGB tactile image. The Apriltag attached to the gripper is solely employed for calibration purposes during the quantitative evaluation in Section 5.4.2 and is not utilized as input to the system. The framework also takes as input the desired goal configuration and estimation priors (Figure 5.2b):

- **Desired Goal Configuration**: A combination of the manipulation objectives discussed in Section 5.3.1.

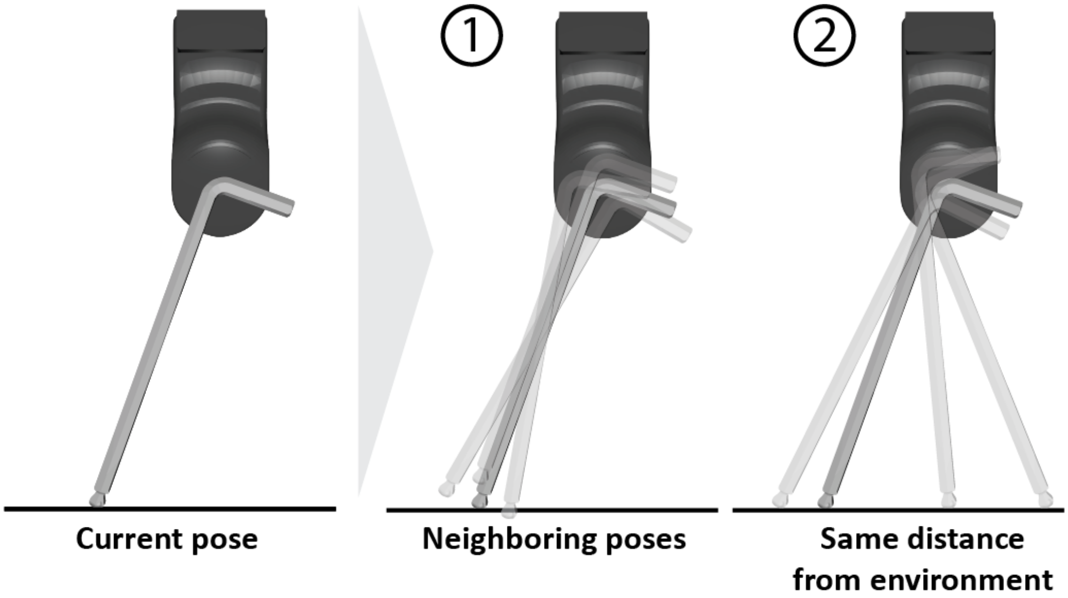


Figure 5.4: Sample set of allowable transitions on Allen key. The object relative to the gripper finger at the current timestep is shown at left. Possible transitions to new poses at the next timestep are shown at right and center. Transitions favored by the first transition likelihoods (neighboring poses) are shown at center, while those favored by the second transition likelihoods (same distance from the ground) are shown at right.

- **Physics Parameter Priors**: The friction parameters at both the intrinsic contact (gripper/object) and the extrinsic contact (object/environment). These priors do not need to be accurate and are manually specified based on physical intuition.
- **Environment Priors**: The orientation and height of the environment in the world frame.

Utilizing these inputs, our **simultaneous tactile estimator-controller** (Figure 5.2c) calculates pose estimates for the object, along with a motion plan to achieve the manipulation objectives (Figure 5.2d). This updated motion plan guides the robot's motion. The framework comprises two main components: **discrete pose estimator** and **continuous pose estimator-controller**, which are described in the next subsections.

5.3.3 Discrete Pose Estimator

The discrete pose estimator computes a probability distribution within a discretized grid of relative gripper/object poses. We first describe the process to create pose distributions from single tactile images, and then how to filter through streams of theses distribution estimates.

The individual tactile images are processed as in Tac2Pose [19]. We first reconstruct a binary mask over the region of contact from raw RGB tactile images using a pixel-to-pixel convolutional neural network (CNN) model as described in [19]. Subsequently, the binary mask is channeled into the Tac2Pose estimator [19], which generates a distribution over possible object poses from a single contact mask.

The Tac2Pose estimator is trained per-object in simulation with rendered contact masks, then transferred directly to the real world. The process for rendering contact masks given an object CAD model is described in detail in [19]. We design a domain randomization procedure tailored for tactile images to ease sim-to-real transfer. These include randomly removing border pixels, tilting the object into and out of the plane of the sensor, randomizing the penetration depth, and randomly removing a fraction of the bottom portion of the sensor (to simulate finger flexing that often occurs during grasping). Once trained, Tac2Pose estimator can run at approximately 50Hz.

We then merge the stream of tactile information with the environment prior via discrete filtering, yielding a filtered probability distribution of the relative object pose. We implement the discrete filter with PGMax [120], running parallel belief propagation for a number of iterations corresponding to the number of variable nodes in the discrete graph. This procedure includes (with some redundant computation) the same belief propagation steps as the Viterbi algorithm [121], a standard algorithm for discrete filtering. Since the computation time is driven by the number of discrete nodes, we marginalize out previous variables each time we incorporate a new observation, maintaining a graph that contains only two nodes. We discretize the pose space by specifying a set of grasp approach directions (normal direction of the grasp surface) relative to the object, then sampling grasps on the object with 5mm of translational resolution, and 10of rotational resolution. The discretized state space consists of 5k-9k poses, depending on the object size. The inference step takes 2-6 seconds per iteration, yielding a slow and coarse but global object pose signal.

Figure 5.3a provides insight into the architecture of the Viterbi algorithm. The variable $\mathcal{X} \in SE(2)$ represents the relative pose between the gripper and the grasped object. At the initial timestep, the environment prior is introduced. Given our prior knowledge of the environment's orientation and height, we can, for each discrete relative object pose within the grid, ascertain which point of the object would be in closest proximity to the environment and compute the corresponding distance. To do so, we transform the object pointcloud (obtained by sampling the object CAD model) by each of the poses in the grid, then save the distance of the closest point in the posed pointcloud to the ground plane in the contact normal direction. The integration of the environment prior involves the multiplication of a Gaussian function over these distances:

$$\mu(\mathcal{X}_0) = P_{\text{Tac2Pose}}(\mathcal{X}_0|I_0, w_0)P_{env}(\mathcal{X}_0|g_0, c^*) \tag{5.1}$$

$$P_{env}(\mathcal{X}_0|g_0, c^*) = \mathcal{N}(p^*_{\text{closest}}(\mathcal{X}_0, g_0, c^*) \cdot \hat{n}_{c^*}; 0, \sigma_{env}) \tag{5.2}$$

where $\mu(\mathcal{X}_0)$ is the probability of the relative gripper/object pose $\mathcal{X}_0$, $P_{\text{Tac2Pose}}(\mathcal{X}_0|I_0, w_0)$ is the single-shot estimate of probability distribution given the tactile image observation I_0, and the gripper width w_0. $P_{env}(\mathcal{X}_0|g_0)$ is the Gaussian function given the gripper pose $g_0 \in SE(2)$ and the environment prior $c^* \in SE(2)$ - the x-axis of c^* represents the environment surface. p^*_{closest} represents the closest point on the object's point cloud to the environment surface, given the relative pose $\mathcal{X}$, gripper pose g_0, and environment prior c^*, and $\hat{n}_{c^*}$ represents the unit vector normal to the environment surface. σ_{env} determines the strength of the environment prior. In essence, the environment prior assigns higher probabilities to the relative poses that are predicted to be closer to the environment.

Subsequently, we incorporate the single-shot tactile pose estimation distribution at every n^{th} step of the continuous pose estimator-controller, where n is approximately five (see Figure 5.3a), since the discrete pose estimator runs slower than the continuous pose estimator-controller. Instead of integrating tactile observations at a fixed frequency, we add the next tactile observation as soon as the discrete filter is ready, once the marginalization step to incorporate the previous tactile observation has been completed.

The transition probabilities impose constraints on tactile observations between consecutive time steps in the discrete graph, including:

- *Continuity*: The pose can transition only to neighboring poses on the pose grid to encourage continuity. (Figure 5.4-1)

- *Persistent Contact*: The height of the closest point to the environment remains consistent across time steps due to the flat nature of the environment. This consistency is enforced through the multiplication of a Gaussian function that factors in the height difference. (Figure 5.4-2)

The first transition probability zeros out the likelihood of any transition to a non-neighboring grid point. Because the discretization of pose space is coarse, we assume the object cannot traverse more than one grid point in a single timestep. A set of allowable transitions corresponding to the first transition probability is visualized in Figure 5.4-1.

The second transition probabilities can be mathematically expressed as follows:

$$\begin{aligned} &P(\mathcal{X}_i|\mathcal{X}_{i-1}) = \\ &\quad \mathcal{N}((p^*_{\text{closest}}(\mathcal{X}_i, g_i, c^*) - p^*_{\text{closest}}(\mathcal{X}_{i-1}, g_{i-1}, c^*)) \cdot \hat{n}_{c^*}; 0, \sigma_{trs}) \end{aligned} \tag{5.3}$$

where σ_{trs} determines the strength of this constraint. A set of transitions that are highly likely given the second transition probabilities are visualized in Figure 5.4-2.

Together, they encode the assumption that the object slides continuously within the grasp. This enables the discrete pose estimator to compute and filter the distribution of relative gripper/object poses, taking into account tactile information, robot proprioception, and environmental priors.

5.3.4 Continuous Pose Estimator-Controller

The continuous pose estimator-controller serves a dual purpose: it takes as input the filtered discrete probability distribution of relative gripper/object poses and outputs a continuous pose estimate and an iteratively updated motion plan in a receding horizon fashion. The Incremental Smoothing and Mapping (iSAM) algorithm [58], which is based on the factor graph model [59], [122], serves as the computational backbone of our estimator-controller. We leverage its graph-based flexible formulation to combine estimation and control objectives as part of one single optimization problem.

The factor graph architecture of the continuous pose estimator-controller is illuminated in Figure 5.3b. Noteworthy variables include g_t, o_t, and c_t, each frames in $SE(2)$, representing the gripper pose, object pose, and contact position, respectively. The orientation of c_t is fixed and aligned with the normal direction of the environment. Additionally, $\mathcal{V}$ represents the set of physics parameters:

- Translational-to-rotational friction ratio at the grasp: F_{max}/M_{max}, where F_{max} and M_{max} are the maximum pure force and torque that it can endure before sliding.
- Friction coefficient at the extrinsic contact between the object and the environment: μ_{max}.

A key advantage of using the factor graph to represent the problem is that we can fuse various sources of information by formulating each piece of information as a factor. Subsequently, we can find the state that best explains the information by jointly minimizing the sum of the factor potentials, i.e. energy function. In other words, priors, measurements, kinematic constraints, physics models, and even control objectives can be represented as factors. This allows us to address both estimation and control problems simultaneously by minimizing a single energy function:

$$\begin{aligned} E(x) = & \underbrace{\sum ||F_{\text{prior}}(\mathbf{x}_{\text{prior}})||^2}_{\text{priors}} + \underbrace{\sum ||F_{\text{meas}}(\mathbf{x}_{\text{meas}})||^2}_{\text{measurements}} \\ & + \underbrace{\sum ||F_{\text{cons}}(\mathbf{x}_{\text{cons}})||^2}_{\text{constraints}} + \underbrace{\sum ||F_{\text{model}}(\mathbf{x}_{\text{model}})||^2}_{\text{models}} \\ & + \underbrace{\sum ||F_{\text{obj}}(\mathbf{x}_{\text{obj}})||^2}_{\text{control objectives}} \end{aligned} \quad (5.4)$$

where F_{prior}, F_{meas}, F_{cons}, F_{model}, and $F_{\text{objective}}$ are the factor potentials associated with priors, measurements, constraints, models, and control objectives, respectively. It is also noteworthy that each of the square terms is normalized by its corresponding noise model, but it is omitted for brevity. The subsets of state variables related to each factor are denoted as $\mathbf{x}_{\text{prior}}$, $\mathbf{x}_{\text{meas}}$, $\mathbf{x}_{\text{cons}}$, $\mathbf{x}_{\text{model}}$, and $\mathbf{x}_{\text{obj}}$. In Figure 5.3b, each circle represents a state variable, and each dot represents a factor. The connections between variables and factors illustrate their relationships. Notably, factors labeled in red accept input from priors, measurements, or objectives, while those in grey stipulate relations between associated variables without taking any inputs.

The continuous estimator-controller comprises two main sections: the left segment, spanning from the initial time to the current moment t, is dedicated to the **estimation** of the object's pose. This estimation component considers priors, measurements, constraints, and physics models to estimate a smooth trajectory for the object's pose. The right segment, covering the time from t to the control horizon $t + T$, is responsible for devising a motion plan to **control** the system and achieving the manipulation objectives. The control component takes into account constraints, physics models, and control objectives to formulate the motion plan.

In the following sections, we define each factor. The arguments of each factor definition are the variables, priors, and measurements that the factor depends on. The right-hand side specifies the quantity we are trying to optimize.

Priors First, the environment (contact) prior is established at the initial time step:

$$F_{cp}(c_1; c^*) = c^{*-1} c_1, \tag{5.5}$$

Here, $c^* \in SE(2)$ contains prior information about the environment's orientation and height. In essence, this factor penalizes the difference between the prior and the estimation. While we employ the logarithm map from the SE(2) Lie Group representation to the se(2) Lie Algebra representation to formulate the output as a three-dimensional vector, we omit the notation for brevity. This abbreviation also applies to other factors where the SE(2) transformation serves as the output.

Additionally, physics priors are imposed by formulating the factor that penalizes the difference between the prior and the estimation:

$$F_{vp}(\mathcal{V}; \mathcal{V}^*) = \mathcal{V} - \mathcal{V}^*. \tag{5.6}$$

where $\mathcal{V}^*$ is the prior for the physics parameters.

Measurements The gripper pose measurement from forward kinematics (g_i^*) is imposed by formulating factor as difference between the measured and the estimated gripper pose:

$$F_{gp}(g_i; g_i^*) := g_i^{*-1} g_i \tag{5.7}$$

The factor graph also takes filtered pose estimations from the discrete pose estimator:

$$F_{tac}(g_i, o_i; \mathcal{X}_{i,\text{MAP}}) = \mathcal{X}_{i,\text{MAP}}^{-1}(g_i^{-1} o_i), \tag{5.8}$$

where $\mathcal{X}_{i,\text{MAP}}$ denotes the filtered maximum a posteriori (MAP) discrete relative pose, and $(g_i^{-1} o_i)$ denotes the continuous estimate of the gripper/object relative pose. Given the higher operating speed of the continuous pose estimator-controller (0.1∼0.2 seconds per iteration) compared to the discrete pose estimator (2∼6 seconds per iteration), the discrete pose estimation factor is integrated when an update is available every few steps within the continuous estimator-controller. This is why we see, in Figure 5.3, that this factor is not imposed at every time step.

Kinematic Constraints Since we assume a flat environment, the location of contact on the environment should not change in the direction perpendicular to the environment surface. Additionally, the change in the tangential direction should be small, given our assumption of quasistatic motion and the absence of abrupt sliding on the environmental surface. We enforce this constraint by formulating a factor and assigning a strong noise model in the perpendicular direction and a relatively weaker noise model in the tangential direction:

$$F_{cc}(c_{i-1}, c_i) = c_{i-1}^{-1} c_i \tag{5.9}$$

Furthermore, we assume we have 3D shape models of the objects and a prior knowledge of the normal direction of the environment. Therefore, as in the discrete filtering step, we can anticipate which part of the object would be in contact with the environment — specifically, the closest point to the environment. Consequently, we introduce a factor that incorporates the distance between the current estimated location of the contact and the closest point of the object to the environment:

$$F_{oc}(o_i, c_i) = p_{\text{closest}}(o_i, c_i) \tag{5.10}$$

where $p_{\text{closest}}(o_i, c_i)$ represents the point in the object's point cloud that is closest to the environment direction, expressed in the contact frame c_i.

Physics Model We impose a friction model based on the limit-surface model [105], [123] as a transition model to capture the dynamics of sliding (F_{fric}). This model provides a relation between the kinetic friction wrench and the direction of sliding at the grasp. In essence, it serves as a guide for predicting how the object will slide in response to a given gripper motion and extrinsic contact location. The relation is formally represented as follows:

$$[\omega, v_x, v_y] \propto [\frac{M}{M_{max}^2}, \frac{F_x}{F_{max}^2}, \frac{F_y}{F_{max}^2}]. \tag{5.11}$$

Here, $[\omega, v_x, v_y]$ denotes the relative object twist in the gripper's frame, i.e. sliding direction, while $[M, F_x, F_y]$ signifies the friction wrench at the grasp. To fully capture the friction dynamics, additional kinematic and mechanical constraints at the extrinsic contact are also considered. These constraints are formulated as follows:

$$M\hat{z} - \vec{l}_{gc} \times \vec{F} = 0, \tag{5.12}$$

$$v_{c,N}(g_{i-1}, o_{i-1}, c_{i-1}, g_i, o_i) = 0, \tag{5.13}$$

$$v_{c,T}(g_{i-1}, o_{i-1}, c_{i-1}, g_i, o_i) = 0 \tag{5.14}$$

$$\perp (F_T = -\mu_{max} F_N \text{ OR } F_T = \mu_{max} F_N), \tag{5.15}$$

In these equations, $\vec{l}_{gc}$ is the vector from the gripper to the contact point, and $v_{c,N}$ and $v_{c,T}$ represent the local velocities of the object at the point of contact in the directions that are normal and tangential to the environment, respectively. F_N and F_T denote the normal and tangential components of the force. Eq. 5.12 specifies that no net torque should be present at the point of extrinsic contact since we are assuming point contact. Eq. 5.13 dictates that the normal component of the local velocity at the point of extrinsic contact must be zero as long as contact is maintained. Eq. 5.14 and Eq. 5.15 work complementarily to stipulate that the tangential component of the local velocity at the contact point must be zero (Eq. 5.14), except in cases where the contact is sliding. In such instances, the contact force must lie on the boundary of the friction cone (Eq. 5.15). By combining Eq. 5.11∼5.15, we establish a fully determined forward model for the contact and object poses, which allows the object pose at step i to be expressed as a function of its previous poses, the current gripper pose, and the physics parameters:

$$o_i^* = f(g_{i-1}, o_{i-1}, c_{i-1}, g_i, \mathcal{V}) \tag{5.16}$$

This relationship can thus be encapsulated as a friction factor:

$$F_{fric}(g_{i-1}, o_{i-1}, c_{i-1}, g_i, o_i, \mathcal{V}) = o_i^{*-1} o_i. \tag{5.17}$$

With all the previously introduced factors combined, the estimation component formulates a smooth object pose trajectory that takes into account priors, tactile measurements, robot kinematics, and physics model.

Control Objective The control segment incorporates multiple auxiliary factors to facilitate the specification of regrasping objectives. First, the desired goal configuration is imposed at the end of the control horizon (F_{goal}). This comprises three distinct sub-factors, corresponding to the three manipulation objectives described in Section 5.3.1, which can be turned on or off, depending on the desired configuration:

1. $F_{\text{goal,go}}$ regulates the desired gripper/object relative pose at o_{t+T} and g_{t+T}.

2. $F_{\text{goal,o}}$ enforces the object's orientation within the world frame at o_{t+T}.

3. $F_{\text{goal,c}}$ dictates the desired contact point at c_{t+T}, thereby facilitating controlled sliding interactions with the environment.

These sub-factors are mathematically expressed as follows:

$$F_{goal,go}(g_{t+T}, o_{t+T}) = p^{g\ -1}_{o,goal}(g^{-1}_{t+T} o_{t+T}), \tag{5.18}$$

$$F_{goal,o}(o_{t+T}) = o^{-1}_{goal} o_{t+T}, \tag{5.19}$$

$$F_{goal,c}(c_{t+T}) = c^{-1}_{goal} c_{t+T}. \tag{5.20}$$

Here, $p^g_{o,goal}$ signifies the target relative gripper/object pose, o_{goal} represents the desired object orientation in the world frame, and c_{goal} is the intended contact point.

Additionally, the F_{motion} factor minimizes the gripper motion across consecutive time steps to reduce redundant motion and optimize for a smooth gripper trajectory.

$$F_{motion}(g_{i-1}, g_i) = g^{-1}_{i-1} g_i \tag{5.21}$$

Concurrently, a contact maintenance factor, F_{cm}, serves as a soft constraint to direct the gripper's motion in a way that prevents it from losing contact with the environment:

$$F_{cm}(g_{i-1}, c_{i-1}, g_i; \epsilon_i) = max(0, \zeta_i(g_{i-1}, c_{i-1}, g_i) + \epsilon_i), \tag{5.22}$$

where ζ_i represents the normal component of the virtual local displacement from step $i-1$ to i at the contact point, assuming the grasp is fixed. The term ϵ_i is a small positive scalar, encouraging ζ_i to be negative, thus fostering a motion that pushes against the environment.

Taken together, these factors cohesively formulate a motion plan from g_{t+1} to g_{t+T}, which is then communicated to the robot. The robot continues to follow the interpolated trajectory of this motion plan until it receives the next update, akin to model predictive control.

5.3.5 Ablation Models

We implemented five ablation models to investigate the contribution of specific components to estimation accuracy:

- Environment priors (height/orientation)
- Multi-shot filtering (v.s. single-shot estimate)
- Continuous estimation (v.s. discrete estimate)
- Quality of contact patch reconstruction
- Accuracy of the physics prior

SS (w/o Env.): This model, equivalent to the previous Tac2Pose algorithm [19], uses a single tactile image and the gripper width to compute the probability distribution of relative poses between the gripper and the grasped object ('Single-shot Tactile Pose Estimate' in Figure 5.2).

SS (w/ Env.): In addition to the tactile image and gripper width, this model incorporates priors on the height and orientation of the environment floor. Comparing this model with 'SS (w/o Env.)' provides insights into the contribution of the environment priors to estimation accuracy. All subsequent ablation models incorporate environment priors.

Filtered (Discrete): This model utilizes the discrete filter to fuse a stream of multiple tactile images ('Filtered Tactile Pose' in Figure 5.2). Comparing with 'SS (w/ Env.)' helps assess the impact of fusing multiple tactile images on accuracy compared to using just a single tactile image.

Discrete+Continuous (Ours): Our proposed model. The following two ablation models leverage privileged information to evaluate potential improvements in estimation accuracy.

Discrete+Continuous (Privileged): This model uses privileged information to synthesize the binary contact patch. From the Apriltag attached to the grasped object, it computes the ground truth relative pose between the gripper and the object. Based on the relative pose, it synthesizes the anticipated binary contact patch rather than inferring it from actual tactile images. Since the same contact patch synthesis method was used during the training of the Tac2Pose model, this model shows how the system would perform if the binary contact patch reconstruction were the same as the ground truth.

Discrete+Continuous (Simulation): This model provides insights into the system's performance under the assumption of an exact physics prior. The methodology involves simulating the object trajectory based on the identical gripper trajectory used in other models. The object trajectory simulation employs a modified factor graph. By imposing only the priors, kinematic constraints, gripper motion, and the physics model, we can find the object trajectory that exactly aligns with the physics model. Consequently, using the same physics parameters for this simulation factor graph as those in our prior and synthesizing the contact patch corresponding to the simulated object trajectory allows us to evaluate how effectively our system would perform with both the exact physics model and contact patch reconstruction.

5.4 Experiments and Results

We conducted a series of experiments on four distinct 3D-printed objects and three household items (illustrated in Figure 5.5) to validate the efficacy of our algorithm. The experiments were designed to:

1. Quantitatively evaluate the algorithm's performance across a variety of target configurations.

2. Qualitatively demonstrate the utility of the algorithm with household items in various

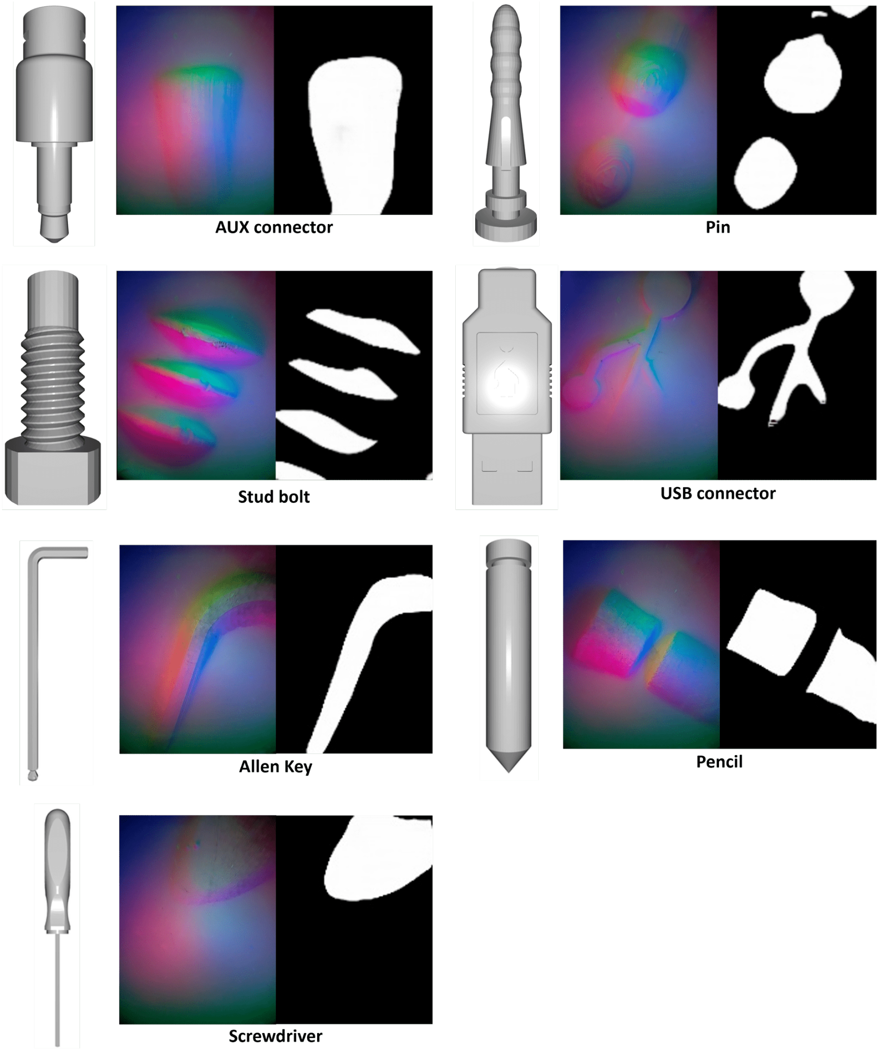


Figure 5.5: Test objects with example tactile images and contact patch reconstruction.

scenarios.

3. Assess the algorithm's applicability to specific real-world tasks, such as object insertion.

5.4.1 Experimental Setup

Figure 5.6 shows the hardware setup, which includes a 6-DoF ABB IRB 120 robot arm, Weiss WSG-50 parallel gripper, and a GelSlim 3.0 sensor [27]. On the table, there is a stage that serves as a flat environment, as well as objects and holes for the insertion experiment. Additionally, there is an Intel RealSense camera used to track the object's pose through

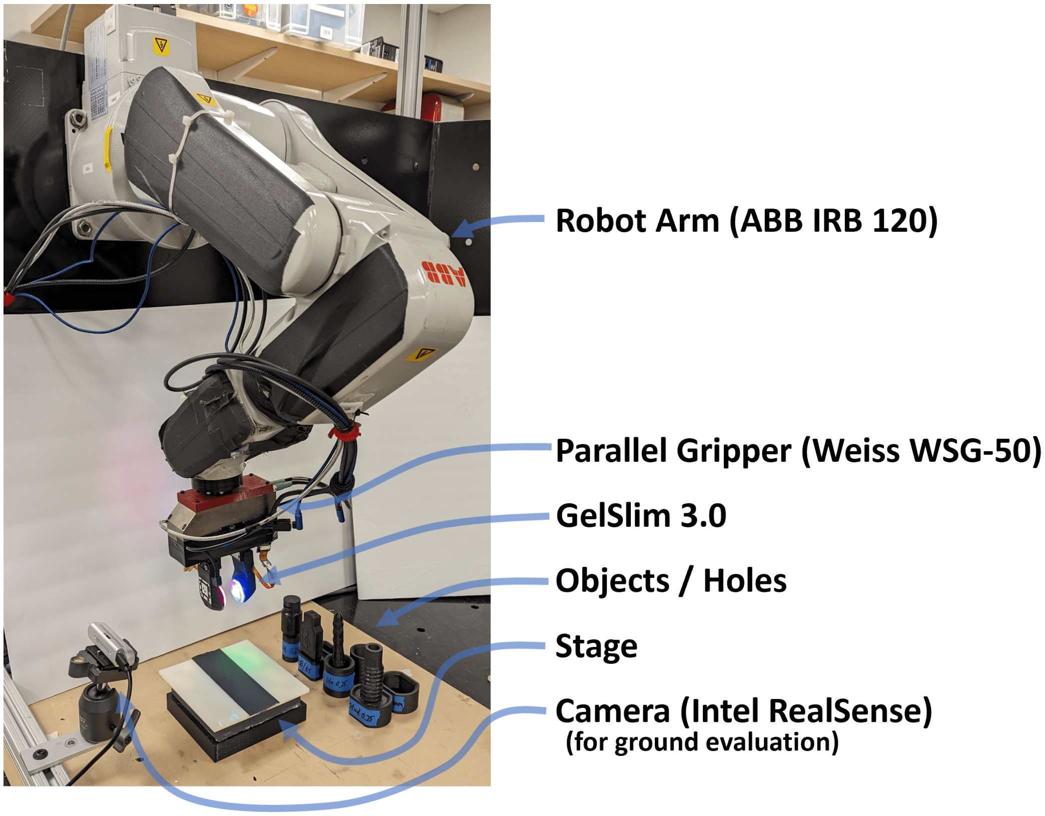


Figure 5.6: Hardware setup

Apriltags attached to the objects to obtain the ground truth pose. The Apriltag attached to the gripper serves the purpose of calibration.

5.4.2 Performance Across Various Goal Configurations

We assessed our algorithm's performance using a total of 18 diverse goal configurations. Our framework allows for specifying goals relative to the gripper (regrasping) and relative to the world frame (reorienting), facilitating different downstream tasks. For example, regrasping can improve grasp stability, enable tactile exploration, and establish a grasp optimized for both force execution and the avoidance of collisions or kinematic singularities in downstream tasks. On the other hand, reorienting the object can enable mating with target objects in the environment or prevent collisions with obstacles. The configurations we evaluate fall into four distinct categories:

Table 5.1: Median Normalized Estimation Errors

	AUX (6 traj.)	Pin (5 traj.)	Stud (3 traj.)	USB (4 traj.)	**Overall** (18 traj.)
SS (w/o Env.)	0.92	2.00	1.47	1.12	1.41
SS (w/ Env.)	0.21	0.12	0.30	0.25	0.20
Filtered (Discrete)	0.17	0.10	0.18	0.17	0.15
Discrete+Continuous (Ours)	**0.10**	**0.07**	**0.07**	**0.07**	**0.07**
Discrete+Continuous (Privileged)	0.06	0.09	0.08	0.07	0.07
Discrete+Continuous (Simulation)	0.03	0.05	0.10	0.06	0.05

- Relative Orientation + Stationary Extrinsic Contact
- Relative Orientation/Translation + Stationary Extrinsic Contact
- Relative Orientation + Global Orientation + Stationary Extrinsic Contact
- Relative Orientation + Sliding Extrinsic Contact

Examples of these four goal configuration types are illustrated in Figure 5.7, along with corresponding plots showcasing estimation accuracy. The red silhouettes that move along with the gripper represent the desired relative pose between the gripper and the object. Conversely, the grey silhouettes depict object poses as measured by Apriltags, which we use as the ground truth object pose. The red dots mark the desired extrinsic contact location. In Figure 5.7c, the other red silhouette signifies the desired object orientation in the global frame. The time series plots on the right column indicate the performance of the proposed and ablation models. These results attest to the algorithm's adeptness in attaining desired goal configurations while showing better estimation performance compared to ablation models.

A summary of each algorithm's estimation performance is presented in Figure 5.8; the error per-object is broken out in Table 5.1. The error values denote the normalized estimation error, computed as follows:

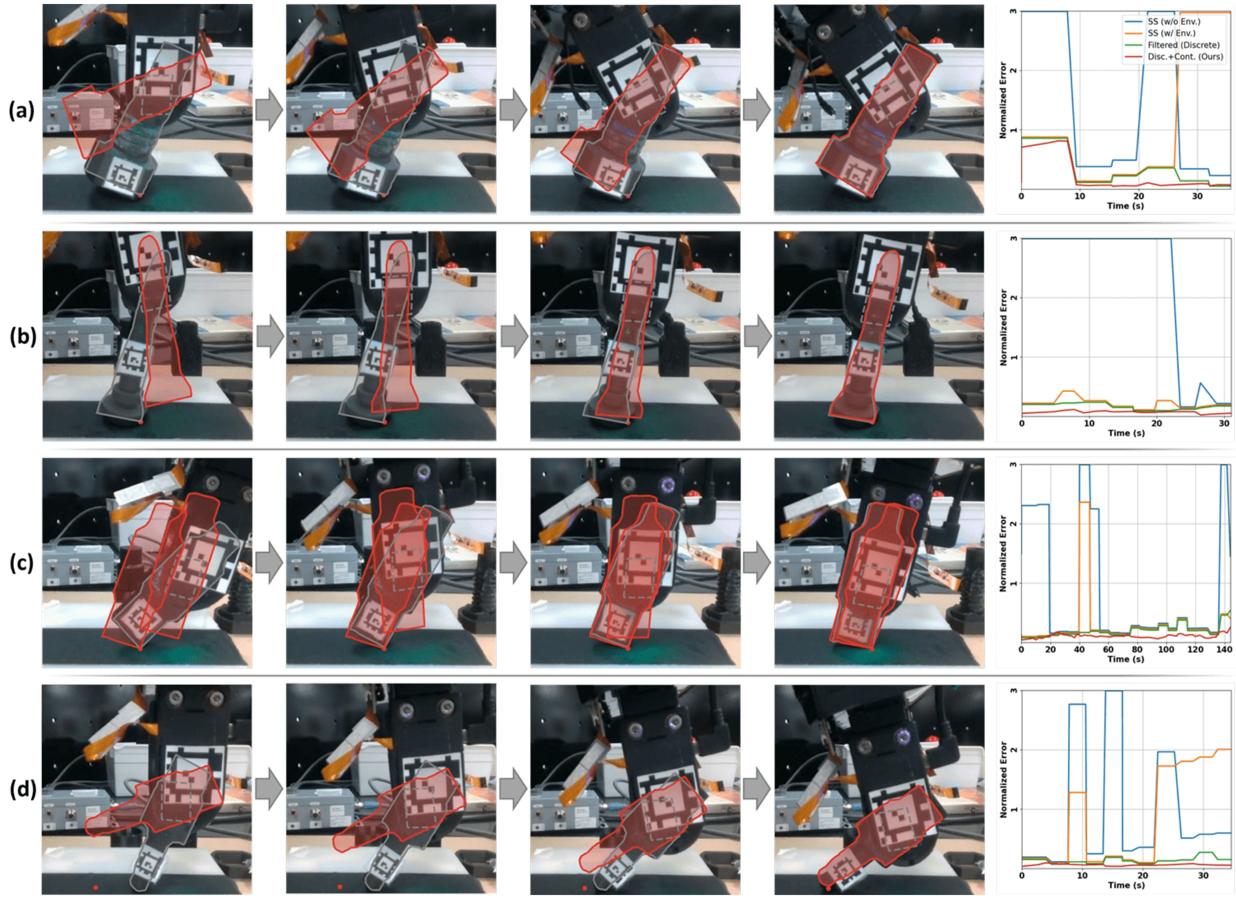


Figure 5.7: Demonstrations of four types of goal configurations: (a) Relative Orientation + Stationary Extrinsic Contact, (b) Relative Orientation/Translation + Stationary Extrinsic Contact, (c) Relative Orientation + Global Orientation + Stationary Extrinsic Contact, and (d) Relative Orientation + Sliding Extrinsic Contact. The right column depicts normalized estimation accuracy for the proposed method and ablation models.

$$\epsilon_{\text{norm}} = ||(\epsilon_{\text{rot}}, \epsilon_{\text{trn}}/(l_{\text{obj}}/2)||_1 \tag{5.23}$$

Here, $||\cdot||_1$ signifies the L1-norm, ϵ_{rot} indicates rotation error in radians, ϵ_{trn} denotes translation error, and l_{obj} represents the object's length. In essence, this value signifies the overall amount of estimation error normalized by objects' size. This analysis reveals how much each system component contributes to the estimation accuracy.

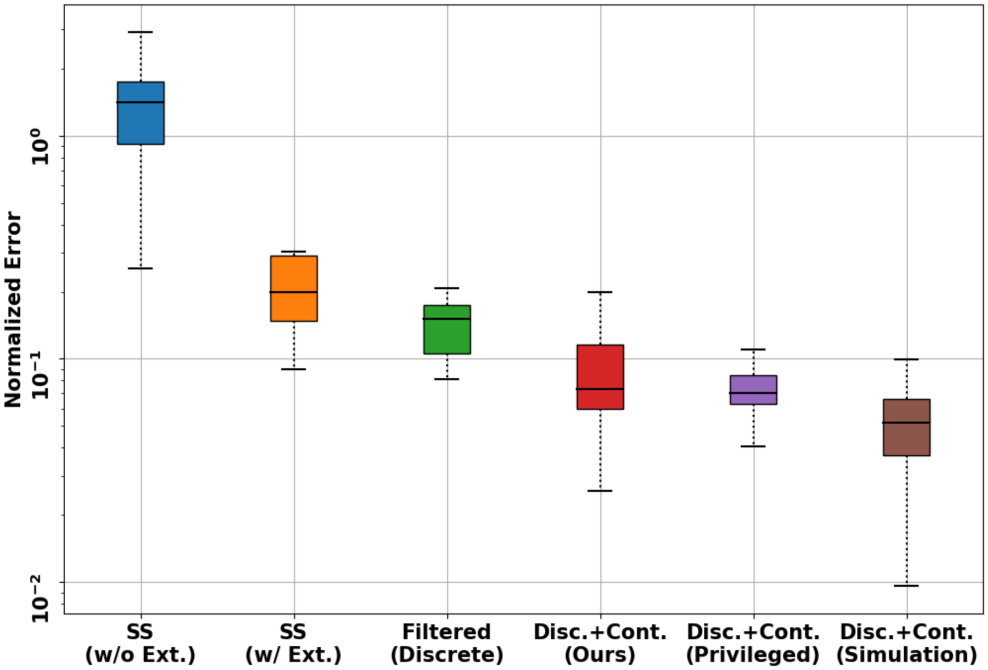


Figure 5.8: Normalized Estimation Errors.

Effect of Environment Priors

Firstly, there is a substantial decrease in normalized error from 1.41 to 0.20 when transitioning from 'SS (w/o Env.)' to 'SS (w/ Env.)'. Without environment priors – no information about the height and orientation of the environment – the estimator suffers due to ambiguity in tactile images, as thoroughly explored in [19]. For most grasps of the objects we experiment with, a single tactile imprint is not sufficient to uniquely localize the object. For instance, the local shape of the pin and the stud exhibits symmetry, making it challenging to distinguish if the object is held upside-down, resulting in a very high estimation error with the 'SS (w/o Env.)' model. In contrast, the 'SS (w/ Env.)' model was able to significantly resolve this ambiguity by incorporating information about the environment.

This suggests that knowing when object is in contact with a known environment can be used effectively to collapse the ambiguity in a single tactile imprint. Although this knowledge, on its own, is a weak signal of pose, it provides global context that, when paired with a tactile imprint, can yield accurate pose estimation. Much prior work prefers vision as a modality to provide global pose context; this analysis demonstrates that prior knowledge of the object and environment (when available) can be leveraged to provide global context instead of introducing additional sensors and algorithms.

Effect of Multi-shot Filtering

Between the 'SS (w/ Env.)' and the 'Filtered (Discrete)' models, the normalized error significantly decreases from 0.20 to 0.15. This shows that fusing a stream of multiple tactile images is effective in improving the estimation accuracy. The discrete filter is able to reduce ambiguity by fusing information over a sequence of tactile images, obtained by traversing the object surface and therefore exposing the estimator to a more complete view of the object geometry. Fusing information over multiple tactile images also robustifies the estimate against noise in the reconstruction of any individual contact mask. The difference is distinctive in the time plots of the normalized error in Figure 5.7. While the 'SS (w/ Env.)' and the 'Filtered (Discrete)' model have an overlapping error profile for the majority of the time, there is a significant amount of portion where the errors of the 'SS (w/ Env.)' model suddenly surges. This is because the 'SS (w/ Env.)' model only depends on a single tactile image snapshot, and therefore does not consider the smoothness of the object pose trajectory over time. In contrast, the error profile of the 'Filtered (Discrete)' model is smoother since it considers consistency in the object pose.

Effect of Continuous Estimation

The median normalized error also decreases from 0.15 of the 'Filtered (Discrete)' model to 0.07 of the 'Discrete+Continuous (Ours)' model. This improvement is attributed to the continuous factor graph refining the discrete filtered estimation with more information in both spatial and temporal resolution. While the discrete filter runs at a lower frequency, the continuous factor graph operates at a higher frequency. This means that it takes in gripper pose measurements even when the discrete estimate from the tactile image is not ready. Additionally, it considers the physics model when computing the estimate. Consequently, the 'Discrete+Continuous (Ours)' model results in a smoother and more physically realistic trajectory estimate, as evident in the error time plots in Figure 5.7.

Potential Effect of Ground Truth Contact Patch Reconstruction

Figure 5.8 suggests that the difference between 'Discrete+Continuous (Ours)' and 'Discrete+Continuous (Privileged)' is not significant. This implies that having the ground truth contact patch reconstruction would not significantly improve the accuracy of the estimation. It suggests that the contact patch reconstruction has sufficiently good quality, retaining significant information compared to the ground truth contact patch. This is attributed to the significant domain randomization incorporated into the contact patch reconstruction during Tac2Pose model training.

When training the Tac2Pose model, the input is the binary contact patch, and the output is the probability distribution of contact poses. The training data for the binary contact patch are synthesized using the local 3D shape of the object model. To overcome the sim-to-real gap in contact patch reconstruction, random errors are intentionally introduced to the synthesized contact patch as discussed in Section 5.3.3. This result indicates that, thanks to effective domain randomization, the model does not suffer significantly from the sim-to-real gap in contact patch reconstruction.

Potential Effect of the Exact Physics Model

Figure 5.8 shows a significant decrease in normalized error when using the simulated physics that exactly aligns with our physics prior. A noteworthy observation is that it does not reduce the normalized error to zero, indicating that our estimation would still not be perfect even with the exact physics model. This aligns with intuition, as tactile observation is a local

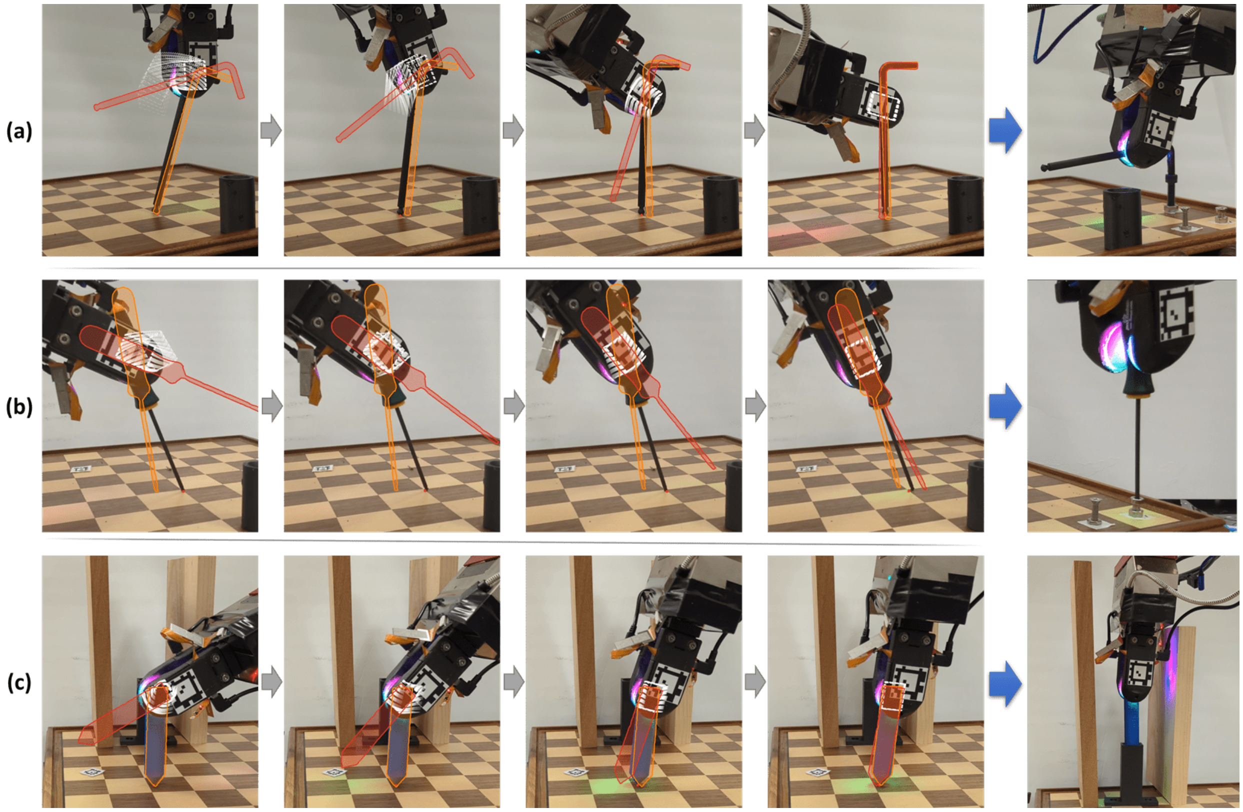


Figure 5.9: Demonstrations with household items in various scenarios: (a) Adjusting the grasp of the Allen key to exert a sufficient amount of torque when screwing a bolt, (b) Adjusting the grasp of the screwdriver to prevent hitting the robot's motion limit while screwing a bolt, (c) Adjusting the grasp of the pencil to ensure the robot does not collide with obstacles when placing the pencil in the pencil holder.

observation and cannot guarantee full observability even when we know the physics exactly.

5.4.3 Demonstration with Household Items in Various Scenarios

We additionally demonstrate our algorithm with real objects in realistic scenarios that we would face in daily life (Figure 5.9):

- *Allen Key*: Adjusting the grasp of the Allen key to exert a sufficient amount of torque when screwing a bolt.
- *Screwdriver*: Adjusting the grasp of the screwdriver to prevent hitting the robot's motion limit or singularity while screwing a bolt.

- *Pencil*: Adjusting the grasp of the pencil to ensure the robot does not collide with obstacles when placing the pencil in the pencil holder.

The left four columns of Figure 5.9 are the snapshots of the motions over time. The rightmost column of the figure illustrates the downstream tasks after the regrasp is done. In the figure, the red silhouettes illustrate the relative goal grasp, manually selected based on the downstream task we want to achieve. The orange silhouettes represent the current estimate of the object pose. The white superimposed rectangles illustrate the planned motion trajectories of the gripper to achieve the desired configurations.

Allen Key (Figure 5.9a)

In the Allen Key example, the initial grasp is on the corner part of the object, making it challenging to exert a sufficient amount of torque. Therefore, we adjusted the grasp by imposing the goal grasp pose on the longer side of the Allen Key. This allows for a longer torque arm length, ensuring the robot can exert a sufficient amount of torque. The orientation of the goal grasp was also set to keep the robot's motion within the feasible range during the screwing process.

A notable observation is that the algorithm first attempts to pivot the object before pushing it down against the floor to slide the grasp. This suggests that the algorithm effectively considers the physics model both on the finger and the floor to plan for a reasonable and intuitive motion. Without incorporating such a physics model, the motion could result in counterintuitive movements, potentially causing the object to slip on the floor.

Screwdriver (Figure 5.9b)

In the Screwdriver example, the initial grasp is configured such that the screwing axis and the robot wrist axis are not aligned. This configuration could lead to issues when attempting to screw at a large angle, as it requires a more extensive motion of the robot arm compared to when the screwing axis and the wrist axis are aligned. Additionally, it may cause the

Table 5.2: Insertion Experiment Results (Success/Attempt)

Clearance	AUX	Pin	Stud	USB
1 mm	10 / 10	6 / 10	7 / 10	7 / 10
0.5 mm	9 / 10	3 / 10	5 / 10	6 / 10

robot arm to reach its motion limit. Conversely, by regrasping the screwdriver and aligning the screwing axis with the wrist axis, the robot can easily screw the bolt with primarily wrist axis rotation. Therefore, we set the goal grasp pose to align the screwing axis and the wrist axis.

While the algorithm was able to get close to the goal grasp, the estimation error was significantly larger than in the other two cases. This is because the screwdriver has less distinctive tactile features than the other items. In Figure 5.5, we can see that the screwdriver has an oval-shaped contact patch without straight lines or sharp corners. In contrast, the Allen key and the pencil exhibit more distinctive straight lines and sharp corners. Since these distinctive features are crucial for resolving estimation uncertainty, the screwdriver shows less estimation accuracy than the other two.

Pencil (Figure 5.9c)

In the pencil example, the robot wrist axis and the pencil are not aligned in the initial grasp. Given the obstacles next to the pencil holder, the robot would likely collide with the obstacles without adjusting the grasp. Therefore, we set the goal grasp to enable the robot to avoid collisions with the obstacles. Consequently, the robot achieved an appropriate grasp while keeping track of the object pose estimate, and then successfully placing the pencil in the holder without colliding with obstacles.

5.4.4 Practical Application: Insertion Task

To validate our algorithm's practical utility, we applied it to a specific downstream task — object insertion with small clearance (1∼0.5 mm). For these experiments, we sampled random goal configurations from the first category (adjusting relative orientation) described in Section 5.4.2. Following this, we aimed to insert the grasped object into holes with 1 mm and 0.5 mm total clearance in diameter.

Table 5.2 summarizes the outcomes of these insertion attempts. The AUX connector, which features a tapered profile at the tip, had a success rate exceeding 90%. On the other hand, the success rate dropped considerably for objects with untapered profiles, especially when the clearance was narrowed from 1 mm to 0.5 mm. The varying performance is consistent with our expectations, given that the algorithm's median normalized pose estimation error is 0.07, which corresponds to approximately 2∼3 mm of translation error as quantified in Section 5.4.2.

These findings indicate that our algorithm is useful in tasks that necessitate regrasping and reorienting objects to fulfill downstream objectives by meeting the goal configuration. However, for applications requiring sub-millimeter accuracy, the algorithm's performance would benefit from integration with a compliant controlled insertion policy (e.g., [15], [18], [55], [124]).

5.5 Conclusion

This paper introduces a novel simultaneous tactile estimator-controller tailored for in-hand object manipulation. The framework harnesses extrinsic dexterity to regrasp a grasped object while simultaneously estimating object poses. This innovation holds particular promise in scenarios necessitating object or grasp reorientation for tasks like insertion or tool use, particularly in cases where the precise visual perception of the object's global pose is difficult due to occlusions.

We show the capability of our algorithm to autonomously generate motion plans for diverse goal configurations that encompass a range of manipulation objectives, then execute them precisely via high-accuracy tactile pose estimation (approximately 2∼3mm of error in median) and closed-loop control. We further demonstrate the practical utility of our approach in solving high-tolerance insertion tasks, as well as showcase our method's ability to generalize to household objects in realistic scenarios, encompassing a variety of material, inertial, and frictional properties.

In future research, our focus will extend to investigating methodologies for autonomously determining optimal target configurations for task execution, eliminating the need for manual specification. Additionally, we are keen on exploring the potential of inferring physics parameters online or integrating a more advanced physics model capable of reasoning about the intricacies of real-world physics.

Chapter 6

Conclusion

6.1 Key Findings

At the beginning of this thesis, we highlighted three major challenges (Interpretability, Observability, and Uncertainty). We formulated four consecutive questions, one in each previous publication, in pursuit of addressing these challenges. In this section, we summarize our responses to these questions and discuss their relevance to the challenges we mentioned.

- ***Is Tactile Sensing Actually Useful?*** → **Yes!**

  Tactile sensing proves more useful than other modalities, such as F/T sensing, especially in tasks involving subtle contact interactions. For example, in the peg-in-hole insertion task, tactile sensing provides intuition on aggregate force and torque on the wrist, as well as subtle variation in relative pose between the gripper and the grasped object, resulting in superior performance compared to F/T sensing.

- ***How Can We Exploit Tactile Sensing More Efficiently?*** → **Use Extrinsic Contact!**

  Extrinsic contact serves as a more efficient, interpretable, and scalable representation of the contact state than raw tactile measurements. It preserves essential information about the contact state with a simple representation, facilitating easier and faster training for manipulation policies. Additionally, the simpler and more interpretable representation makes simulation more straightforward than raw tactile measurements, alleviating the challenge of collecting data through real experiments.

- ***How Can We Reason About Extrinsic Contact Interaction With Tactile Sensing?*** → **Simultaneously Estimate and Control Contact State!**

  To address the challenges of reasoning about extrinsic contact interaction, we developed a method that simultaneously estimates and controls the contact state. This approach involves incorporating tactile feedback, kinematics, and physical models to navigate the uncertainty introduced by physical interactions, providing an accurate estimation of the contact states and enhancing the robot's ability to interact with its environment.

- ***How Can We Achieve Extrinsic Dexterity With Tactile Sensing?*** → **Integrate Object/Physics Model!**

  In pursuit of achieving extrinsic dexterity, we incorporated an object model and a physics model of friction dynamics, addressing intrinsic and extrinsic contacts, into our framework. This integration enables the robot to estimate the object's pose and associated contact configurations. By simultaneously controlling these elements, the robot gains the capability to perform precise in-hand sliding regrasps by pushing the object against the environment, showcasing extrinsic dexterity. This effective utilization of tactile sensing enhances the robot's overall dexterity.

Based on the insights gained from addressing the questions above, we outline strategies to tackle the major challenges in tactile sensing and achieve extrinsic dexterity:

- Interpretability: Leveraging interpretable abstractions, such as extrinsic contact information, provides a pathway to enhance understanding and efficient utilization of tactile feedback. By focusing on key features like contact location, contact forces, and direction/magnitude of sensor deformation, we establish a more interpretable representation, facilitating effective integration into robotic systems.

- Observability: The fusion of tactile feedback, kinematics, and physical models over time enhances observability, allowing for a more comprehensive understanding of the system's state. Integrating additional information, such as vision or force/torque sensing, may further contribute to improved system observability, enabling robots to reason about contact interactions occurring beyond the direct tactile sensing locations.

- Uncertainty: Simultaneous estimation and control of the contact state offers a robust approach to address uncertainty introduced by physical interactions. By actively managing uncertainty during exploratory robot motions, the system can maintain accurate estimates of contact states, providing a foundation for reliable tactile feedback utilization. Additionally, integrating object models and friction dynamics models contributes to uncertainty reduction by providing a contextual understanding of the grasped object and its interaction with the environment when leveraging extrinsic dexterity.

6.2 Limitation and Future Work

6.2.1 Manipulating Through Unstructured Environment

While the current research excels in controlled and structured environments, particularly showcasing its capabilities in scenarios like flat floors with minimal clutter, it does not directly address the intricacies of manipulating through unstructured environments, such as deformable or cluttered surroundings. The challenges posed by unpredictable and dynamic settings remain open questions within the scope of this work. Future investigations should explore learning-based methods or adaptive strategies to enable robotic manipulation in unstructured environments, paving the way for real-world applicability and versatility.

6.2.2 Generalizing Over Different Tactile Sensors

While the proposed methods showcase the efficacy of specific tactile sensing modalities, the challenge lies in creating a unified protocol for interpreting tactile data across various sensor types. Future research should explore methodologies to establish a standardized approach for interpreting tactile information, enabling seamless integration and generalization across different tactile sensors.

6.2.3 Speeding Up the Motion

The current study prioritizes the successful execution of tasks with a focus on precision and control. However, the speed of robotic motion during manipulation emerges as a crucial aspect warranting further exploration. Notably, the demo videos within this thesis are accelerated between 5 to 10 times the actual speed. Our unpublished preliminary investigation suggests that sensor latency may be a bottleneck affecting the speed. In this initial exploration, we measured an approximate latency of 50 milliseconds from sensor stimulation to the computer receiving the tactile image. This latency introduces a gap of about 5 millimeters when aiming for motions at a speed of 1 m/s, a factor that can be critical for tasks demanding sub-millimeter accuracy.

While latency can vary based on hardware specifications or signal processing protocols, the present study does not specifically address minimizing latency or precisely compensating for it by introducing additional latency in other sensing modalities like robot proprioception. Future research endeavors should explore techniques, particularly those aimed at reducing or managing tactile sensor latency, to enhance the optimization and speed of robotic motions without compromising precision.

6.2.4 Automating the Goal Specification

While the current research contributes to enhancing tactile sensing for specific tasks, it does not explicitly address the challenge of automating the goal specification process. In the current framework, defining goals for robotic manipulation is typically a manual or predefined process. Future work should explore avenues for automating this aspect, integrating planning algorithms that enable robots to autonomously infer and adapt to task objectives. This would mark a significant step toward making robotic manipulation systems more autonomous and adaptable to various scenarios.

References

[1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, “Mastering the game of go with deep neural networks and tree search,” *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[2] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, “Learning robust perceptive locomotion for quadrupedal robots in the wild,” *Science Robotics*, vol. 7, no. 62, eabk2822, 2022.

[3] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, “Anymal parkour: Learning agile navigation for quadrupedal robots,” *arXiv preprint arXiv:2306.14874*, 2023.

[4] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, “Champion-level drone racing using deep reinforcement learning,” *Nature*, vol. 620, no. 7976, pp. 982–987, 2023.

[5] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, *et al.*, “Rt-2: Vision-language-action models transfer web knowledge to robotic control,” in *Conference on Robot Learning*, PMLR, 2023, pp. 2165–2183.

[6] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, “Tidybot: Personalized robot assistance with large language models,” *Autonomous Robots*, 2023.

[7] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence, “Interactive language: Talking to robots in real time,” *IEEE Robotics and Automation Letters*, 2023.

[8] Y. Narang, K. Storey, I. Akinola, M. Macklin, P. Reist, L. Wawrzyniak, Y. Guo, A. Moravanszky, G. State, M. Lu, *et al.*, “Factory: Fast contact for robotic assembly,” in *Robotics: Science and Systems (RSS)*, 2022.

[9] B. Tang, M. A. Lin, I. Akinola, A. Handa, G. S. Sukhatme, F. Ramos, D. Fox, and Y. Narang, “Industreal: Transferring contact-rich assembly tasks from simulation to reality,” in *Robotics: Science and Systems (RSS)*, 2023.

[10] T. Pang, H. T. Suh, L. Yang, and R. Tedrake, “Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models,” *IEEE Transactions on Robotics*, 2023.

[11] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” in *Robotics: Science and Systems (RSS)*, 2023.

[12] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, “Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots,” *arXiv preprint arXiv:2402.10329*, 2024.

[13] Z. Fu, T. Z. Zhao, and C. Finn, “Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation,” in *arXiv*, 2024.

[14] W. Yuan, S. Dong, and E. H. Adelson, “Gelsight: High-resolution robot tactile sensors for estimating geometry and force,” *Sensors*, vol. 17, no. 12, p. 2762, 2017.

[15] S. Kim and A. Rodriguez, “Active extrinsic contact sensing: Application to general peg-in-hole insertion,” in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 10 241–10 247.

[16] S. Kim, D. K. Jha, D. Romeres, P. Patre, and A. Rodriguez, “Simultaneous tactile estimation and control of extrinsic contact,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 563–12 569. DOI: 10.1109/ICRA48891.2023.10161158.

[17] A. Bronars, S. Kim, P. Patre, and A. Rodriguez, “TEXterity: Tactile Extrinsic deXterity,” *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[18] S. Dong, D. K. Jha, D. Romeres, S. Kim, D. Nikovski, and A. Rodriguez, “Tactile-RL for insertion: Generalization to objects of unknown geometry,” in *2021 International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 6437–6443.

[19] M. Bauza, A. Bronars, and A. Rodriguez, “Tac2pose: Tactile object pose estimation from the first touch,” *The International Journal of Robotics Research*, vol. 42, no. 13, pp. 1185–1209, 2023.

[20] M. T. Mason, “Toward robotic manipulation,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 1–28, 2018.

[21] K.-T. Yu and A. Rodriguez, “Realtime state estimation with tactile and visual sensing for inserting a suction-held object,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 1628–1635.

[22] A. Bicchi, J. K. Salisbury, and D. L. Brock, “Contact Sensing from Force Measurements,” *The International Journal of Robotics Research*, vol. 12, no. 3, p. 249, 1993.

[23] S. Li, S. Lyu, J. Trinkle, and W. Burgard, “A comparative study of contact models for contact-aware state estimation,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[24] L. Manuelli and R. Tedrake, “Localizing external contact using proprioceptive sensors: The contact particle filter,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[25] D. Ma, S. Dong, and A. Rodriguez, "Extrinsic contact sensing with relative-motion tracking from distributed tactile measurements," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 11 262–11 268.

[26] S. Dong and A. Rodriguez, "Tactile-based insertion for dense box-packing," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 7953–7960.

[27] I. H. Taylor, S. Dong, and A. Rodriguez, "Gelslim 3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger," pp. 10 781–10 787, 2022.

[28] D. E. Whitney, "Quasi-static assembly of compliantly supported rigid parts," *Journal of Dynamic Systems, Measurement, and Control*, vol. 104, no. 1, pp. 65–77, Mar. 1982, ISSN: 0022-0434. DOI: 10.1115/1.3149634. eprint: https://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/104/1/65/5777806/65_1.pdf.

[29] D. Whitney and J. Nevins, "Research on advanced assembly automation," *Computer*, vol. 10, no. 12, pp. 24–38, Dec. 1977, ISSN: 1558-0814. DOI: 10.1109/C-M.1977.217597.

[30] H. Bruyninckx, S. Dutre, and J. De Schutter, "Peg-on-hole: A model based solution to peg and hole alignment," in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, IEEE, vol. 2, 1995, pp. 1919–1924.

[31] M. E. Came, T. Lozano-Pérez, and W. P. Seering, "Assembly strategies for chamferless parts," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1989, pp. 472–477.

[32] W. S. Newman, Y. Zhao, and Y. H. Pao, "Interpretation of force and moment signals for compliant peg-in-hole assembly," in *IEEE International Conference on Robotics and Automation*, IEEE, vol. 1, 2001, pp. 571–576. DOI: 10.1109/robot.2001.932611.

[33] V. Gullapalli, "Learning control under extreme uncertainty," in *Advances in neural information processing systems*, 1993, pp. 327–334.

[34] S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search," in *2015 International Conference on Robotics and Automation (ICRA)*, IEEE, 2015.

[35] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, and A. M. Agogino, "Deep reinforcement learning for robotic assembly of mixed deformable and rigid objects," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 2062–2069.

[36] G. Schoettler, A. Nair, J. Luo, S. Bahl, J. A. Ojea, E. Solowjow, and S. Levine, "Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[37] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks," in *International Conference on Robotics and Automation (ICRA)*, 2019.

[38] Y. Liu, D. Romeres, D. K. Jha, and D. Nikovski, "Understanding multi-modal perception using behavioral cloning for peg-in-a-hole insertion tasks," *RSS 2020 workshop on Advances Challenges in Imitation Learning for Robotics*, 2020.

[39] S. Dong, D. Ma, E. Donlon, and A. Rodriguez, "Maintaining grasps within slipping bound by monitoring incipient slip," in *International Conference on Robotics and Automation (ICRA)*, 2018.

[40] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning*, PMLR, 2018, pp. 1587–1596.

[41] R. D. Howe, "Tactile sensing and control of robotic manipulation," *Advanced Robotics*, vol. 8, no. 3, pp. 245–261, 1993.

[42] A. Rodriguez, "The unstable queen: Uncertainty, mechanics, and tactile feedback," *Science Robotics*, vol. 6, no. 54, 2021. DOI: 10.1126/scirobotics.abi4667.

[43] M. Kaess, A. Ranganathan, and F. Dellaert, "Isam: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[44] K.-.-T. Yu and A. Rodriguez, "Realtime state estimation with tactile and visual sensing. application to planar manipulation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 7778–7785.

[45] E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, and A. Rodriguez, "Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 1927–1934.

[46] M. Bauza, O. Canal, and A. Rodriguez, "Tactile mapping and localization from high-resolution tactile imprints," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 3811–3817.

[47] M. Bauza, E. Valls, B. Lim, T. Sechopoulos, and A. Rodriguez, "Tactile object pose estimation from the first touch with geometric contact rendering," in *Conference on Robot Learning (CoRL)*, IEEE, 2020.

[48] D. Ma, E. Donlon, S. Dong, and A. Rodriguez, "Dense tactile force estimation using gelslim and inverse fem," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 5418–5424.

[49] P. Sodhi, M. Kaess, M. Mukadam, and S. Anderson, "Learning tactile models for factor graph-based estimation," in *2021 International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 13 686–13 692.

[50] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. 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*, vol. 5, no. 3, pp. 3838–3845, 2020.

[51] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 698–700, 1987.

[52] S. Dong, D. Ma, E. Donlon, and A. Rodriguez, "Maintaining grasps within slipping bounds by monitoring incipient slip," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 3818–3824.

[53] F. R. Hogan, J. Ballester, S. Dong, and A. Rodriguez, "Tactile dexterity: Manipulation primitives with tactile feedback," in *2020 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2020, pp. 8863–8869.

[54] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," *The International Journal of Robotics Research*, 2021.

[55] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, and R. Tachibana, "Deep reinforcement learning for high precision assembly tasks," in *IEEE International Conference on Intelligent Robots and Systems*, IEEE, Institute of Electrical and Electronics Engineers Inc., Dec. 2017, pp. 819–825, ISBN: 9781538626825. DOI: 10.1109/IROS.2017.8202244.

[56] M. A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Learning multimodal representations for contact-rich tasks," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 582–596, 2020.

[57] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.

[58] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "Isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[59] F. Dellaert, "Factor graphs and gtsam: A hands-on introduction," Georgia Institute of Technology, Tech. Rep., 2012, p. 4.

[60] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the brownian motion," *Phys. Rev.*, vol. 36, pp. 823–841, 5 Sep. 1930. DOI: 10.1103/PhysRev.36.823.

[61] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[62] S. Wang, M. Lambeta, P.-W. Chou, and R. Calandra, "Tacto: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3930–3937, 2022.

[63] F. Veiga, H. Van Hoof, J. Peters, and T. Hermans, "Stabilizing novel objects by learning to predict tactile slip," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, 2015, pp. 5065–5072.

[64] M. Li, Y. Bekiroglu, D. Kragic, and A. Billard, "Learning of grasp adaptation through experience and tactile sensing," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Ieee, 2014, pp. 3339–3346.

[65] S. Tian, F. Ebert, D. Jayaraman, M. Mudigonda, C. Finn, R. Calandra, and S. Levine, "Manipulation by feel: Touch-based control with deep predictive models," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 818–824.

[66] R. Kolamuri, Z. Si, Y. Zhang, A. Agarwal, and W. Yuan, "Improving grasp stability with rotation measurement from tactile sensing," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 6809–6816.

[67] N. Doshi, O. Taylor, and A. Rodriguez, "Manipulation of unknown objects via contact configuration regulation," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 2693–2699. DOI: 10.1109/ICRA46639.2022.9811713.

[68] S. Kim and A. Rodriguez, "Active extrinsic contact sensing: Application to general peg-in-hole insertion," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 241–10 247. DOI: 10.1109/ICRA46639.2022.9812017.

[69] F. Dellaert, "Factor graphs: Exploiting structure in robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 141–166, 2021.

[70] S. Yang, G. Chen, Y. Zhang, H. Choset, and F. Dellaert, "Equality constrained linear optimal control with factor graphs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 9717–9723.

[71] J. Dong, M. Mukadam, F. Dellaert, and B. Boots, "Motion planning as probabilistic inference using gaussian processes and factor graphs.," in *Robotics: Science and Systems*, vol. 12, 2016.

[72] M. Mukadam, J. Dong, F. Dellaert, and B. Boots, "Steap: Simultaneous trajectory estimation and planning," *Autonomous Robots*, vol. 43, no. 2, pp. 415–434, 2019.

[73] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," vol. 32, 2017.

[74] M. Bauza, F. R. Hogan, and A. Rodriguez, "A data-efficient approach to precise and controlled pushing," in *Conference on Robot Learning*, PMLR, 2018, pp. 336–345.

[75] I. Mordatch, Z. Popović, and E. Todorov, "Contact-invariant optimization for hand manipulation," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, 2012, pp. 137–144.

[76] B. Sundaralingam and T. Hermans, “Geometric in-hand regrasp planning: Alternating optimization of finger gaits and in-grasp manipulation,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 231–238.

[77] Y. Hou, Z. Jia, and M. T. Mason, “Fast planning for 3D any-pose-reorienting using pivoting,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 1631–1638.

[78] J. Shi, J. Z. Woodruff, P. B. Umbanhowar, and K. M. Lynch, “Dynamic in-hand sliding manipulation,” *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 778–795, 2017.

[79] B. Sundaralingam and T. Hermans, “Relaxed-rigidity constraints: Kinematic trajectory optimization and collision avoidance for in-grasp manipulation,” *Autonomous Robots*, vol. 43, pp. 469–483, 2019.

[80] T. Chen, J. Xu, and P. Agrawal, “A system for general in-hand object re-orientation,” in *Conference on Robot Learning*, PMLR, 2022, pp. 297–307.

[81] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, “Learning complex dexterous manipulation with deep reinforcement learning and demonstrations,” in *Proceedings of Robotics: Science and Systems (RSS)*, 2017.

[82] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal, “Visual dexterity: In-hand dexterous manipulation from depth,” in *Icml workshop on new frontiers in learning, control, and dynamical systems*, 2023.

[83] A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. Van Wyk, A. Zhurkevich, B. Sundaralingam, *et al.*, “Dextreme: Transfer of agile in-hand manipulation from simulation to reality,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 5977–5984.

[84] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, “Learning dexterous in-hand manipulation,” *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[85] W. Huang, I. Mordatch, P. Abbeel, and D. Pathak, “Generalization in dexterous manipulation via geometry-aware multi-task learning,” *arXiv preprint arXiv:2111.03062*, 2021.

[86] S. Pai, T. Chen, M. Tippur, E. Adelson, A. Gupta, and P. Agrawal, “Tactofind: A tactile only system for object retrieval,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 8025–8032. DOI: 10.1109/ICRA48891.2023.10160289.

[87] S. Luo, W. Yuan, E. Adelson, A. G. Cohn, and R. Fuentes, "Vitac: Feature sharing between vision and tactile sensing for cloth texture recognition," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 2722–2727.

[88] C. Higuera, S. Dong, B. Boots, and M. Mukadam, "Neural contact fields: Tracking extrinsic contact with tactile sensing," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 12 576–12 582.

[89] Y. Shirai, D. K. Jha, A. U. Raghunathan, and D. Hong, "Tactile tool manipulation," in *2023 International Conference on Robotics and Automation (ICRA)*, IEEE, 2023.

[90] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1385–1401, 2021.

[91] N. Sunil, S. Wang, Y. She, E. Adelson, and A. Rodriguez, "Visuotactile affordances for cloth manipulation with local control," in *Conference on Robot Learning*, PMLR, 2023, pp. 1596–1606.

[92] N. C. Dafle, A. Rodriguez, R. Paolini, B. Tang, S. S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrigge, "Extrinsic dexterity: In-hand manipulation with external forces," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 1578–1585.

[93] P. Sodhi, M. Kaess, M. Mukadanr, and S. Anderson, "Patchgraph: In-hand tactile tracking with learned surface normals," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 2164–2170.

[94] J. Zhao, M. Bauza, and E. H. Adelson, "FingerSLAM: Closed-loop unknown object localization and reconstruction from visuo-tactile feedback," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 8033–8039. DOI: 10.1109/ICRA48891.2023.10161489.

[95] S. Suresh, M. Bauza, K.-T. Yu, J. G. Mangelson, A. Rodriguez, and M. Kaess, "Tactile SLAM: Real-time inference of shape and pose from planar pushing," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 11 322–11 328.

[96] R. Li, R. Platt, W. Yuan, A. ten Pas, N. Roscup, M. A. Srinivasan, and E. Adelson, "Localization and manipulation of small parts using gelsight tactile sensing," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, IEEE, 2014, pp. 3988–3993.

[97] M. Bauza, A. Bronars, Y. Hou, I. Taylor, N. Chavan-Dafle, and A. Rodriguez, "SimPLE: A visuotactile method learned in simulation to precisely pick, localize, regrasp, and place objects," *arXiv preprint arXiv:2307.13133*, 2023.

[98] S. Dikhale, K. Patel, D. Dhingra, I. Naramura, A. Hayashi, S. Iba, and N. Jamali, "Visuotactile 6d pose estimation of an in-hand object using vision and tactile sensor data," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2148–2155, 2022.

[99] G. Izatt, G. Mirano, E. Adelson, and R. Tedrake, "Tracking objects with point clouds from vision and touch," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, IEEE, 2017, pp. 4000–4007.

[100] T. Anzai and K. Takahashi, "Deep gated multi-modal learning: In-hand object pose changes estimation using tactile and image data," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 9361–9368.

[101] S. Suresh, Z. Si, S. Anderson, M. Kaess, and M. Mukadam, "Midastouch: Monte-carlo inference over distributions across sliding touch," in *Conference on Robot Learning*, PMLR, 2023, pp. 319–331.

[102] T. Kelestemur, R. Platt, and T. Padir, "Tactile pose estimation and policy learning for unknown object manipulation," *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2022.

[103] Y. Hou, Z. Jia, and M. Mason, "Manipulation with shared grasping," in *Robotics: Science and Systems (RSS)*, 2020.

[104] J. Shi, H. Weng, and K. M. Lynch, "In-hand sliding regrasp with spring-sliding compliance and external constraints," *IEEE Access*, vol. 8, pp. 88 729–88 744, 2020.

[105] N. Chavan-Dafle, R. Holladay, and A. Rodriguez, "In-hand manipulation via motion cones," in *Robotics: Science and Systems (RSS)*, 2018.

[106] N. Chavan-Dafle and A. Rodriguez, "Prehensile pushing: In-hand manipulation with push-primitives," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2015, pp. 6215–6222.

[107] A. Nagabandi, K. Konolige, S. Levine, and V. Kumar, "Deep dynamics models for learning dexterous manipulation," in *Conference on Robot Learning*, PMLR, 2020, pp. 1101–1112.

[108] V. Kumar, E. Todorov, and S. Levine, "Optimal control with learned local models: Application to dexterous manipulation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 378–383.

[109] M. Lepert, C. Pan, S. Yuan, R. Antonova, and J. Bohg, "In-hand manipulation of unknown objects with tactile sensing for insertion," in *Embracing Contacts-Workshop at ICRA 2023*, 2023.

[110] J. Pitz, L. Röstel, L. Sievers, and B. Bäuml, "Dextrous tactile in-hand manipulation using a modular reinforcement learning architecture," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1852–1858. DOI: 10.1109/ICRA48891.2023.10160756.

[111] H. Van Hoof, T. Hermans, G. Neumann, and J. Peters, "Learning robot in-hand manipulation with tactile features," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, IEEE, 2015, pp. 121–127.

[112] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik, "General in-hand object rotation with vision and touch," in *Conference on Robot Learning*, PMLR, 2023, pp. 2549–2564.

[113] G. Khandate, S. Shang, E. T. Chang, T. L. Saidi, J. Adams, and M. Ciocarlie, "Sampling-based exploration for reinforcement learning of dexterous manipulation," in *Robotics: Science and Systems (RSS)*, 2023.

[114] Z.-H. Yin, B. Huang, Y. Qin, Q. Chen, and X. Wang, "Rotating without seeing: Towards in-hand dexterity through touch," in *Robotics: Science and Systems (RSS)*, 2023.

[115] L. Sievers, J. Pitz, and B. Bäuml, "Learning purely tactile in-hand manipulation with a torque-controlled hand," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 2745–2751.

[116] Y. Yuan, H. Che, Y. Qin, B. Huang, Z.-H. Yin, K.-W. Lee, Y. Wu, S.-C. Lim, and X. Wang, "Robot synesthesia: In-hand manipulation with visuotactile sensing," *arXiv preprint arXiv:2312.01853*, 2023.

[117] M. Merwe, Y. Wi, D. Berenson, and N. Fazeli, "Integrated object deformation and contact patch estimation from visuo-tactile feedback," in *Robotics: Science and Systems*, 2023.

[118] C. Higuera, J. Ortiz, H. Qi, L. Pineda, B. Boots, and M. Mukadam, "Perceiving extrinsic contacts from touch improves learning insertion policies," *arXiv preprint arXiv:2309.16652*, 2023.

[119] L. Kim, Y. Li, M. Posa, and D. Jayaraman, "Im2contact: Vision-based contact localization without touch or force sensing," in *Conference on Robot Learning*, PMLR, 2023, pp. 1533–1546.

[120] G. Zhou, N. Kumar, A. Dedieu, M. Lázaro-Gredilla, S. Kushagra, and D. George, "Pgmax: Factor graphs for discrete probabilistic graphical models and loopy belief propagation in jax," *arXiv preprint arXiv:2202.04110*, 2022.

[121] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[122] F. Dellaert, M. Kaess, *et al.*, "Factor graphs for robot perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[123] S. Goyal, "Planar sliding of a rigid body with dry friction: Limit surfaces and dynamics of motion," Ph.D. dissertation, Cornell University Ithaca, NY, 1989.

[124] T. Z. Zhao, J. Luo, O. Sushkov, R. Pevceviciute, N. Heess, J. Scholz, S. Schaal, and S. Levine, "Offline meta-reinforcement learning for industrial insertion," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 6386–6393.