Slip-Aware Robotic Manipulation: Leveraging Tactile Sensing for Gripper Control and Optimized Robot Motion

Neel Anand Jawale

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

Master of Science

University of Washington

2024

Committee:

Xu Chen

Santosh Devasia

Steve Brunton

Program Authorized to Offer Degree:
Mechanical Engineering

University of Washington

Abstract

Slip-Aware Robotic Manipulation:
Leveraging Tactile Sensing for Gripper
Control and Optimized Robot Motion

Neel Anand Jawale

Chair of the Supervisory Committee:
Xu Chen
Department of Mechanical Engineering

Ensuring safe and reliable object handling without slippage remains a critical challenge in robotic manipulation, especially as robots are increasingly deployed in industrial applications. Traditional methods often treat slip as a binary event (slip/no-slip); however, accurately quantifying slip as a continuous variable is essential for precise and adaptive control. This continuous measurement allows slip to be integrated as a control variable, enabling strategies such as adjusting gripper force or position and optimizing trajectories to minimize slippage. In this thesis, we leverage tactile sensing to achieve real-time slip detection and quantification. Utilizing machine learning models, we accurately measure slip in real time and incorporate this feedback into sophisticated control algorithms to effectively mitigate slippage. Additionally, we demonstrate a proof-of-concept showing how sampling-based Model Predictive Control can optimize robot motions to identify and execute trajectories with minimal slip.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

MPC: Model Predictive Control

MPPI: Model Predictive Path Integral (Control)

POC: Proof of Concept

PID: Proportional Integral Derivative

RF: Random Forest

GB: Gradient Boosting

ACKNOWLEDGMENTS

This thesis represents the collective efforts, support, and encouragement of many individuals who have contributed to its realization. It is my privilege to acknowledge and thank them here.

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Xu Chen, for his invaluable guidance and unwavering support throughout this project. His mentorship has been instrumental in shaping both this thesis and the underlying research. I also extend my sincere thanks to my committee members, Professor Santosh Devasia and Professor Steve Brunton, for their insightful feedback and constructive suggestions, which enhanced the quality of this work. I am grateful to my project collaborators - Navneet Kaur, Amy Santoso, and Xiaohai Hu, whose contributions were vital to the success of this project. I would also like to thank my colleagues from the MACS Lab, including Aparajit Venkatesh, Thomas Chu, Colin Acton, Arun Nandagopal, Jonas Beachy, and many others, for their constant support and for fostering a collaborative and inspiring work environment. Lastly, I wish to acknowledge my friends, whose unwavering encouragement and companionship have been a source of strength during challenging times. While the list is too extensive to include here, their impact on my journey has been profound and I am deeply thankful for their presence in my life.

To all who have been part of this journey, thank you for your belief in me and for helping me grow, both personally and professionally.

DEDICATION

to my parents, my brother, and my friends

Chapter 1

INTRODUCTION

Grasping is one of the most fundamental yet challenging problems in robotics. Despite its simplicity for humans, who perform it effortlessly with remarkable dexterity, grasping presents significant hurdles for robots due to the inherent uncertainties in perception, control, and interaction with unstructured environments. This difficulty resonates with the perspective of Moravec's Paradox, which emphasizes that tasks perceived as easy for humans due to millions of years of evolutionary refinement, are often the hardest for robots. Robotic grasping requires precise coordination of sensing, planning, and control to handle the complexities arising from varied object geometries, materials, and dynamic interactions. For instance, tactile sensing is essential for grasping and manipulating objects in scenarios where visual feedback is limited, as demonstrated by early research on dexterous robotic hands [4, 5, 18, 39, 41]. These challenges have led to diverse approaches in robotic manipulation, making grasping a cornerstone of robotics research.

Robotic grasping typically involves three stages: pre-grasp, during-grasp, and post-grasp. In the pre-grasp stage, the focus is on planning an appropriate approach trajectory and contact points, often using vision-based perception to estimate object pose and geometry [12, 37, 42, 45]. The during-grasp stage requires the precise application of forces and torques to secure the object, incorporating sensor feedback for stability and robustness [5, 13, 31]. Finally, the post-grasp stage encompasses manipulation tasks, ensuring the object remains stable while being transported or used. While significant advancements have been made in the first two stages, the post-grasp stage remains underexplored, particularly in addressing slip during object handling. Slip, which occurs when the object begins to lose contact stability due to insufficient or misaligned forces, is a critical challenge in dynamic

manipulation scenarios. Effective slip detection and mitigation mechanisms are essential for enabling robust and safe robotic manipulation in real-world applications [36].

Slip detection methods play a pivotal role in improving robotic manipulation, as they allow robots to adapt dynamically to the physical state of an object during manipulation. Traditional approaches to slip detection relied on simple force-torque sensors, which monitored external forces to identify instability [10, 32]. However, these methods often lack the sensitivity required for nuanced detection in dynamic environments. Recent advancements have leveraged high-resolution tactile sensors such as GelSight, which utilize optical marker-based deformation analysis to capture subtle contact dynamics [14, 48]. Machine learning techniques further enhance slip detection capabilities by learning from tactile data and enabling robust generalization across diverse objects and tasks [1, 25]. These advancements underscore the growing importance of tactile sensing in achieving reliable post-grasp manipulation and addressing the persistent challenge of slip. By focusing on this stage, our research aims to bridge the gap in robotic manipulation and improve the robustness of real-world applications.

The goal of this research is rooted in the ambition to apply advanced control strategies to the domain of gripper-object interactions, particularly addressing the challenge of slip during manipulation. Achieving this requires a comprehensive understanding of the phenomenon of slip, extending beyond binary classification to a quantitative representation. Such a representation can be effectively utilized in control algorithms for improving manipulation robustness. This thesis aims to contribute to this domain in the following ways:

- **Understanding Optical Tactile Features**: This work provides an in-depth analysis of the optical tactile feature space to identify features that serve as proxies for surface interactions. These features are derived from optical flow generated by the tactile sensor output, inspired by vector field mathematics.

- **Slip Detection and Severity Estimation Models**: We develop independent models for slip detection and slip severity estimation. These models are evaluated in real-

world manipulation scenarios, demonstrating their effectiveness in characterizing slip dynamics.

- **Integration of Slip Detection and Control**: A framework is proposed that integrates slip detection and slip severity estimation into a gripper control scheme. This integration allows for adaptive and robust manipulation by dynamically responding to slip events.

- **Slip-Aware Manipulation using Trajectory Optimization**: We introduce the concept of minimal slip paths in robotic manipulation and propose a proof-of-concept (POC) implementation of a slip-aware sampling-based Model Predictive Control (MPC) scheme for a pick-and-reach task. The scheme leverages an approximate analytical slip cost, demonstrating the feasibility of incorporating slip awareness into planning and control.

These contributions aim to advance the understanding of slip dynamics and their control in robotic manipulation, providing a foundation for future research in tactile-based manipulation and its integration into sophisticated robotic systems.

Chapter 2

TACTILE FEATURE SYNTHESIS: APPLYING VECTOR FLOW ANALYSIS FOR RICH TACTILE FIELD FEATURES

2.1 Introduction

2.1.1 Tactile Sensing

Tactile sensing research began in the 1970s with piezoelectric elements used as strain sensors [33] and has since expanded to encompass a range of sensor types capable of detecting properties like mass, shape, texture, slip, and hardness. These sensors utilize piezoelectric, capacitive array, optical, and magnetic technologies [44]. Within this spectrum, optical marker-based tactile sensors such as TacTip [46], TouchRoller [8], and GelSight [14,48], stand out for their ability to capture high-resolution surface details. TacTip employs a camera to track pin movements within a biomimetic membrane, responding dynamically to contact. TouchRoller, meanwhile, gathers geometric data as it rolls across a surface. In this research, we focus on the GelSight sensor, which combines a gel-covered elastomer and directional LED lighting to track marker displacements, providing accurate deformation measurements through vector field analysis.

2.1.2 Tactile Feature Synthesis

Tactile sensing has become a critical component in enabling robots to interact effectively within complex, unstructured environments, providing them with the dexterity required for precise manipulation tasks. By mimicking aspects of human touch, tactile sensing enhances robots' abilities to handle delicate objects [21], recognize textures [7], and prevent slip [25], making it an invaluable modality for achieving nuanced interactions. Consequently, the extraction and synthesis of tactile features from raw sensor data are essential for interpreting

this sensory information effectively.

Numerous feature extraction methods have been developed for various types of tactile sensors, each designed to meet specific task demands. Hjorth parameters, for example, were initially designed for electroencephalography (EEG) signal analysis but have been adapted for tactile signal processing to uncover statistical characteristics in tactile data [30]. Voronoi-based feature extraction has also shown effectiveness, using the geometry of Voronoi tessellation to gather information on contact points, tip displacement, and shear direction [11]. This method enables complex insights into pressure distribution and contact interactions between the sensor and object by organizing tactile data into Voronoi diagrams.

Recent advancements in spatio-temporal data representation further extend tactile sensing capabilities. Techniques such as Spatio-Temporal Gradient Representation leverage principles from neuromorphic engineering to transform tactile sensor data into continuous spatio-temporal surfaces [17]. This approach is particularly effective for tracking dynamic touch regions and reducing sensitivity to sensor noise, providing reliable feedback for complex manipulation tasks. Similarly, Graph Convolutional Networks (GCNs) have been developed to interpret spatial relationships in tactile data by representing signals as graph nodes, thus modeling the spatial structure of touch and pressure patterns. [40]

However, these methods often fall short in capturing the intricate surface interactions during object handling in robotic manipulation. In this chapter, we show that well-engineered features from an optical tactile sensor can provide rich insights into these interactions. In the remaining sections of this thesis, we focus specifically on addressing the problem of slip, defined here as the relative motion between a grasped object and the tactile sensor surface. Slip can occur due to several factors, such as insufficient friction between the object and gripper, external forces, or irregularities in the object's surface.

In this research, we introduce vector flow analysis as a comprehensive method for interpreting tactile deformation fields. By examining the direction and magnitude of deformation vectors across the tactile field, vector flow analysis reveals critical contact dynamics vital for tasks like slip detection and force distribution. This method provides a detailed view of

the sensor-object interaction, capturing fine-grained tactile feedback that scalar-based approaches might overlook. In the subsequent chapter, we demonstrate how these features can be used to train machine learning models for slip detection and quantification with high accuracy.

2.1.3 Contributions

- Development of an optical tactile flow field to facilitate feature extraction from the tactile sensor.
- Application of Vector Flow Analysis to engineer features from the tactile deformation vector field.

2.2 Methodology

2.2.1 Tactile Flow Generation

The GelSight Mini sensor is equipped with an internal camera that captures deformation on an elastomeric gel surface as a sequence of images. The gel is embedded with 63 black markers to enable precise pixel tracking. By employing optical flow analysis [43], the deformation field is measured by estimating marker movement across video frames relative to a reference frame, which depicts the gel's undisturbed state. This method provides a detailed view of surface interactions and deformations. An illustration of the displacement markers showing the deformation field is provided in Figure 2.1.

2.2.2 Kinematics of the Tactile Deformation Field

The velocity of the tactile deformation field is key to detecting rapid contact changes that result from the gel's movement on the sensor surface. Velocity represents the rate of change of gel displacement, and the x- and y-components of the velocity for marker i are defined as:

$$v_{x_i} = \frac{\Delta d_{x_i}}{\Delta t}, \quad v_{y_i} = \frac{\Delta d_{y_i}}{\Delta t} \tag{2.1}$$

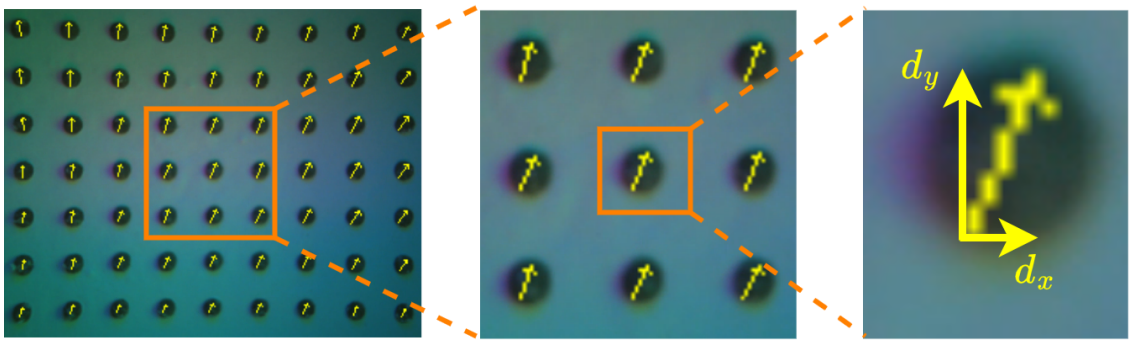

Figure 2.1: The figure shows the deformation vector fields through marker displacement, highlighting the displacement vector components of a specific marker.

where v_{x_i} and v_{y_i} are the velocity components, d_{x_i} and d_{y_i} represent displacement in the x- and y-directions, and $\Delta t (= 0.04\text{s})$ is the sampling time of the GelSight sensor. For the $N (= 63)$ markers on the GelSight mini sensor, the mean velocities are:

$$\bar{v}_x = \frac{1}{N}\sum_{i=1}^{N} v_{x_i}, \quad \bar{v}_y = \frac{1}{N}\sum_{i=1}^{N} v_{y_i} \tag{2.2}$$

To capture the overall motion dynamics, the L_2 norm of the velocity components v_x and v_y is used as a primary feature:

$$\bar{v}_{net} = \sqrt{\bar{v}_x^2 + \bar{v}_y^2} \tag{2.3}$$

2.2.3 Vector Analysis of the Tactile Deformation Field

The deformation vector field provides critical information about contact dynamics and forces [43,50]. We apply vector flow analysis to derive additional features from the deformation field, such as divergence and curl. Divergence represents the net rate of expansion or contraction of the field, often associated with normal force, while curl indicates the presence of rotational forces, as illustrated in Figure 2.2. Rotational forces can increase curl and cause divergence fluctuations, highlighting the interaction between rotational and normal forces on the sensor's

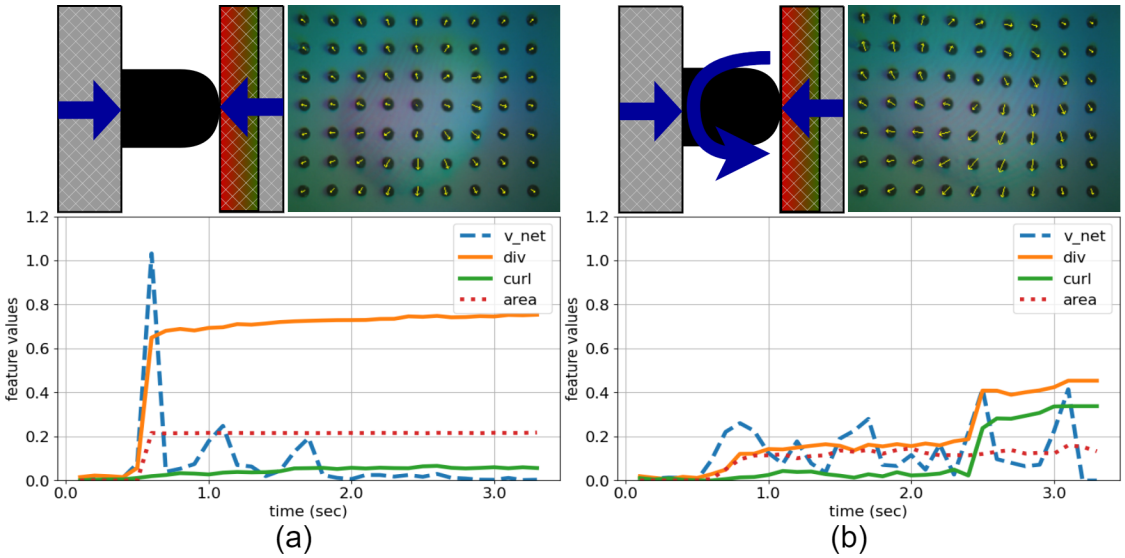

Figure 2.2: Two tactile interaction scenarios are shown: (a) Static grasping, where no relative motion exists between the object and the sensor, showing a noticeable increase in divergence. (b) Object rotation, introducing torsional stress and resulting in increases in both divergence and curl, along with non-uniformity in the contact area, as seen in the graph.

surface. For the discrete deformation field extracted from the tactile sensor, divergence and curl are defined as:

$$\begin{aligned} \nabla \cdot \vec{D} &= \sum_{i=1}^{N} \left(\frac{d_{x_{i+1}} - d_{x_{i-1}}}{2\Delta x} + \frac{d_{y_{i+1}} - d_{y_{i-1}}}{2\Delta y} \right) \\ \nabla \times \vec{D} &= \sum_{i=1}^{N} \left(\frac{d_{y_{i+1}} - d_{y_{i-1}}}{2\Delta x} - \frac{d_{x_{i+1}} - d_{x_{i-1}}}{2\Delta y} \right) \end{aligned} \tag{2.4}$$

where Δx and Δy represent spatial resolution between markers.

The rate of change of these vector quantities captures dynamic tactile events by representing changes in force distribution and directionality at the contact surface. These rates are expressed as:

$$\begin{aligned} \frac{\Delta(\nabla \cdot \vec{D})}{\Delta t} &= \frac{(\nabla \cdot \vec{D})_t - (\nabla \cdot \vec{D})_{t-\Delta t}}{\Delta t} \\ \frac{\Delta(\nabla \times \vec{D})}{\Delta t} &= \frac{(\nabla \times \vec{D})_t - (\nabla \times \vec{D})_{t-\Delta t}}{\Delta t} \end{aligned} \tag{2.5}$$

A rapid change in divergence indicates a possible grip loss due to inadequate normal force, while a spike in curl rate signifies tangential or rotational shifts between the sensor and object.

2.2.4 Normalized Contact Area

The study of slip benefits from analyzing the contact area between an object and the tactile sensor, especially changes over time. Large fluctuations in contact area suggest disturbances at the sensor surface. Rapid changes in area imply higher interaction severity. The left graph in Figure 2.2 illustrates a stable contact area under normal force, indicating a secure grip, while the right graph shows significant variation under combined torsional and normal forces, indicative of complex interactions and slip. This is quantified using the 3D-Recon technique [48], which reconstructs the surface depth of the tactile sensor using a photometric stereo algorithm. Contact area is defined by counting pixels where the depth exceeds a threshold of 1, compensating for minor gel surface variations and ensuring measurement precision. The normalized contact area, A_n, is calculated as:

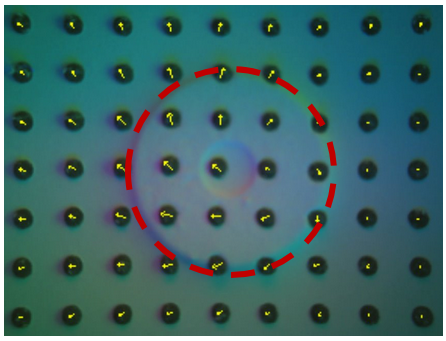

Figure 2.3: The figure highlights the area of contact between the cap of a marker and the tactile sensor.

$$A_n = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} I(z_{ij} > z_t)}{m \times n} \tag{2.6}$$

where I is an indicator function that activates if the depth z_{ij} at any pixel exceeds the threshold z_t, and $m(= 320)$ and $n(= 240)$ are the image dimensions. The rate of change of A_n is then:

$$\frac{\Delta A_n}{\Delta t} = \frac{A_n(t) - A_n(t - \Delta t)}{\Delta t} \tag{2.7}$$

2.2.5 Baseline Features

As noted in [49] and [25], entropy and its rate of change are valuable for slip detection. Entropy of the deformation field, as in these studies, measures the randomness in marker displacement distribution. For a discrete field, entropy can be calculated using Shannon entropy:

$$H(X) = -\sum_{x \in X} p(x) \log p(x) \tag{2.8}$$

where X represents the histogram of marker displacements [16]. Both entropy (H) and its rate of change ($\frac{\Delta H}{\Delta t}$) are used as baseline features, providing a comparison against vector field features in slip detection and severity estimation.

Chapter 3

SLIP DETECTION & SEVERITY ESTIMATION: EMPLOYING MACHINE LEARNING TO IDENTIFY AND QUANTIFY SLIP

3.1 Introduction

Imagine a robot in a warehouse carefully handling delicate objects, needing to ensure that each item remains undamaged. In modern warehouses, robotic manipulators are frequently used for such tasks, but a common issue they face is object slippage. When an item slips from the robot's grasp, it can easily lead to damage, making slip prevention a critical priority. Traditionally, slip is often viewed in simple terms - either it occurs, or it does not. However, this binary perspective limits our ability to implement advanced controls that could more effectively address slip. Moreover, relying only on visual feedback for slip detection has inherent challenges. Visual sensors can be blocked or disrupted by poor lighting or obstacles, and they require external viewpoints, which are not always practical in tight spaces or busy warehouse environments. To address these limitations, this research focuses on enhancing slip detection methods to understand the complexities of slip better as it happens. Additionally, this work also explores ways to quantify slip aiding in formulating responsive control algorithms. This approach could lead to more precise handling of delicate objects, preventing slippage and ensuring safe, efficient robotic manipulation in real-world settings.

3.1.1 Slip Detection

Significant foundational work has demonstrated the effectiveness of tactile data in slip detection [24]. Optical methods have utilized contact surface eccentricity to assess object deformation [27]. The integration of machine learning has further extended slip detection capabilities, with Support Vector Machines (SVMs) applied to TacTip sensor data [28]. Neu-

ral networks have been employed for slip classification [1], and GelSight sensor data has been used to enhance object shape detection [35]. Additionally, entropy-based approaches leverage learning algorithms and shear marker displacement to estimate slip likelihood [25, 49]. This study focuses on extracting tactile features through vector field analysis of the deformation field and utilizing learned models for slip detection.

3.1.2 Continuous Slip Estimation

Research on predicting the relative velocity between grippers and manipulated objects using tactile sensing alone is limited, primarily due to the inherent complexities. One method employs a nonlinear observer with the SUNTouch tactile sensor to estimate sliding velocity [10], but this approach faces challenges when applied to objects with small radii of curvature or high deformability. Another approach leverages linear regression with capacitive-based nib-structure tactile sensors for slip speed prediction [19], showing potential despite limitations in feature space and a low correlation with the regression target. Recent advancements have utilized Convolutional Neural Networks (CNNs) with BioTac sensor data to predict the slipping speed of objects [9]. This method performs effectively for rigid objects but struggles to generalize to deformable items. In this study, we aim to establish a direct relationship between slip velocity and tactile features extracted through vector field analysis, utilizing learned models. By extracting features from optical tactile sensors, we demonstrate that our approach enables the development of models capable of generalizing effectively across various object types. This method also eliminates the need for heuristic-based solutions, addressing the complex interaction dynamics at the sensor surface in a data-driven manner.

3.2 Methodology

3.2.1 Slip Detection Model

Utilizing quantifiable and physical features extracted from the tactile sensor facilitates the application of statistical theory-based algorithms, such as classifiers, in slip detection endeav-

ors [26]. Commonly employed algorithms in this domain include Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Trees (DT), Random Forest (RF), and Linear Discriminant Analysis (LDA). Research has shown that ensemble tree algorithms, particularly when bagged or boosted, are effective in slip detection because of their rapid processing and excellent performance with non-linear data [34]. Hence, our study will concentrate on using Random Forest (RF) [6] and Gradient Boosting (GB) [15] as our slip detection models, evaluating their effectiveness against baseline and newly proposed tactile features. We utilize the Scikit-learn library to implement these classifiers for analysis and inferencing.

3.2.2 Slip Severity Estimation Model

Once a slip detection model is learned and evaluated, we focus on quantifying slip severity. The quantification is based on understanding the physics behind the tactile deformation field and the velocities of objects slipping against sensor surfaces. We observed that a higher velocity of slippage often correlates with an increased difficulty in regaining control, suggesting a higher severity of slip. Our approach seeks to establish a correlation between tactile sensory features and object velocity, enabling the prediction of slip severity using data from tactile sensors alone. This method could be particularly useful in environments where external vision-based systems are not feasible. We employ two deep learning-based models using PyTorch: a Long Short-Term Memory (LSTM) network [23] and a Multilayer Perceptron (MLP) [20]. These models are chosen to explore both the sequential nature of the data and the relationships among static features at discrete time points, respectively. We then compare the performance of these models using baseline and proposed tactile features.

3.3 Experiments & Results

The first experiment in this section focused on deploying the developed slip detection models and evaluating their performance through comparative analysis. The second experiment aimed to implement and assess the slip severity models, using the same flow features extracted from the tactile deformation field as in the slip detection task.

3.3.1 Slip Detection

Experimental Setup and Data Collection

Data collection for slip detection was conducted across three scenarios: static, grasp, and slip, as outlined in Figure 3.1. In the static scenario, the robot, programmed using the MoveIt framework, executed 10 distinct trajectories, each with five progressively tighter grasps. This procedure aimed to capture a range of vector field intensities across the 15 selected objects, maintaining precise spatial poses without inducing slip. The recording duration was standardized across all scenarios and objects to ensure consistency, with each object following identical trajectories. For the grasp scenario, each object was sequentially grasped, with the gripper moving from an initial position p, representing minimal contact, to a final position p+10, pausing for 5 seconds before each position increment. Both static and grasp scenarios were labeled as '0' to indicate the absence of slip. In the slip scenario, slippage was intentionally induced while the objects remained in contact with the GelSight sensors. This setup enabled the collection of data representing varying slip intensities over timeframes identical to those in the no-slip experiments, with this data labeled as '1'. The dataset was manually annotated by human observers to ensure accurate differentiation between static and slipping states. Following

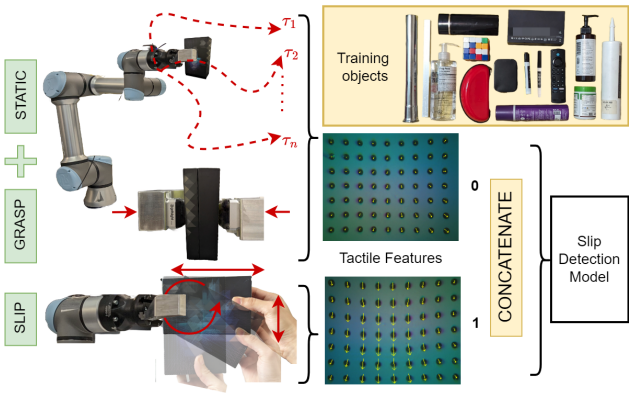

Figure 3.1: Schematic representation of the data collection pipeline for the Slip Detection experiment. The STATIC + GRASP scenario details the methodology for acquiring 'no-slip' data labeled 0, while the SLIP scenario illustrates the process for gathering data indicative of slip labeled 1. Both datasets are combined during the training phase for input into the slip detection model.

data concatenation, the training dataset comprised approximately 66,000 data points. In this dataset, all columns except the last represent feature values, with the final column containing annotations and each row corresponding to an individual data point.

Results

Classification Metrics: To evaluate the classification models' effectiveness comprehensively, a diverse range of metrics was employed, as outlined in Table 3.1. These metrics were designed to capture multiple facets of the models' performance. Specifically, True Positives (TP) represent cases accurately identified as positive, True Negatives (TN) denote cases correctly classified as negative, False Positives (FP) refer to cases mistakenly labeled as positive, and False Negatives (FN) are cases wrongly categorized as negative.

Table 3.1: Classification metrics summary

Metric	**Definition**	**Formula**
Accuracy (A)	Overall correctness	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision (P)	Positives' accuracy	$\frac{TP}{TP+FP}$
Recall (R)	Positive instance identification	$\frac{TP}{TP+FN}$
F1 Score (F)	Precision & recall balance	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}}$

Cross-validation: We performed hyperparameter tuning for RF and GB, specifying values for parameters such as *max_depth*, *max_features*, *min_samples_leaf*, *min_samples_split*, and *n_estimators*. For RF, we set these values to 20, 3, 5, 10, and 40, respectively, while for GB, we assigned the first three parameters to 9, 450, and 115, leaving the rest at their default settings. These models were trained using all feature categories: baseline, proposed, and combined. To ensure fair evaluation across both classes, we used Stratified 5-fold cross-validation. As

Table 3.2: Stratified 5-fold cross-validation results

Metrics		A	P	R	F
RF	Baseline	93.94	94.83	92.31	93.55
	Proposed	99.11	98.76	99.11	98.93
	Combined	**99.37**	**99.1**	**99.38**	**99.24**
GB	Baseline	93.97	94.75	92.46	93.59
	Proposed	99.22	98.97	99.17	99.07
	Combined	**99.57**	**99.49**	**99.47**	**99.48**

shown in Table 3.2, RF and GB achieved mean accuracies of 99.11% and 99.22%, respectively, when trained with the proposed features. Combining these with baseline features slightly improved accuracy, with increases of 0.26% for RF and 0.35% for GB. This suggests that the proposed features effectively capture the relationship with slip occurrences.

Generalization to unseen objects: Both trained classifier models were tested on previously unencountered objects, selected to evaluate their performance across varied shapes and textures: a soft-covered book, a thermocol duster, a smooth wooden plank, a porous sponge, and a pair of scissors with varying surface features. The results, shown in Table 3.3, demonstrate the classifiers' effectiveness in accurately detecting slip instances across the five unseen object datasets. The classifiers achieve an accuracy over 99% for scissors and duster, despite the scissors' abrupt changes in surface area. However, both classifiers encountered difficulty in producing satisfactory results with the sponge. This may be due to the sponge's porous nature, leading to reduced sensor contact, further influenced by its high deformability.

Table 3.3: Results of trained classifiers on unseen objects

Metrics		A	P	R	F
 Book	**RF**	**91.76**	**84.73**	**98.27**	**91.0**
	GB	83.71	83.74	98.96	72.58
 Scissors	RF	99.51	99.28	99.54	99.41
	GB	**99.57**	**99.44**	**99.54**	**99.49**
 Plank	**RF**	**98.11**	**96.55**	**99.17**	**97.84**
	GB	97.77	95.55	99.48	97.47
 Sponge	**RF**	**74.78**	**62.38**	**95.64**	**75.51**
	GB	43.71	41.91	99.64	59.01
 Duster	RF	99.49	99.56	99.28	99.42
	GB	**99.68**	**99.61**	**99.66**	**99.64**

3.3.2 Slip Severity Estimation

Experimental Setup and Data Collection

In this section, we evaluate slip severity under controlled conditions using a dataset comprising 15 objects previously selected for slip detection experiments. The experimental setup, depicted in Figure 3.2, involved securely fixing objects to an elevated platform to ensure a constant position within the world frame. The gripper was actuated vertically along the y-axis at predetermined target velocities, with the gripping force tuned marginally below the slippage threshold of each object. This configuration facilitated controlled slip events during the upward motion of the gripper. The focus was placed on linear slip velocities, excluding angular components, to align with the downstream task of tactile-guided vertical sliding.

Slip velocity, denoted as v_{slip}, represents the rate of object displacement during slippage and was utilized as the ground truth in this study. Data were collected across five velocity profiles: [0.8, 2.3, 3.8, 4.5, 6.7] cm/s, with five repetitions conducted for each velocity per object to ensure statistical robustness. Slip velocities were measured using ArUco markers affixed to the gripper. The data were processed using an Exponential Weighted Moving Average (EWMA) filter to mitigate noise and achieve smoother measurements. The Realsense camera, operating at 60 Hz, required synchronization with the GelSight sensor. To achieve this, the ArUco pose data were downsampled to 25 Hz using the ROS `throttle` node, ensuring alignment between data streams. Instances where the object disengaged from the sensor, ceasing movement, were recorded as a slip velocity of zero, marking the conclusion of the slippage event.

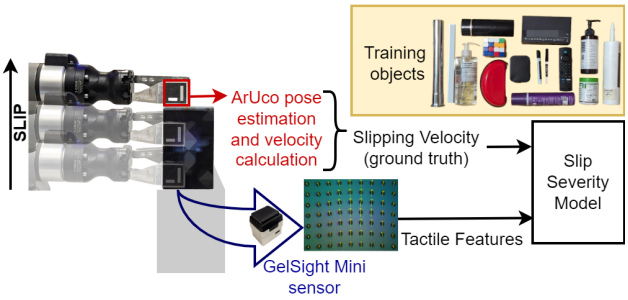

Figure 3.2: Illustration of the data acquisition framework for the Slip Severity Estimation Model. The gripper, equipped with a GelSight Mini sensor, is programmed to slide over a fixed object. This setup synchronously captures tactile feedback and slip velocity data, which is used as ground truth to train the neural network for slip severity estimation.

Results

Evaluation Metrics: To evaluate the performance of our learned slip severity estimation models, we utilized metrics that capture different aspects of predictive accuracy and error magnitude. These metrics are pertinent to estimation analysis, where predictions are continuous and should reflect the true values as closely as possible. Additionally, in the context of time series analysis, these metrics are instrumental in understanding the temporal consistency of predictions. The chosen metrics are summarized in Table 3.4. Here, y_i are the actual values, $\hat{y}_i$ are the predicted values, $\bar{y}$ is the mean of actual values, and n is the number

Table 3.4: Estimation metrics summary

Metric	**Definition**	**Formula**
MAE	Mean absolute diff.	$\frac{1}{n}\sum \lvert y_i - \hat{y}_i \rvert$
RMSE	Square root of MSE	$\sqrt{\frac{1}{n}\sum (y_i - \hat{y}_i)^2}$
R^2	Variance proportion	$1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$

of observations.

Cross-validation: The performance of LSTM and MLP models in estimating slip severity was evaluated using a leave-one-object-out cross-validation methodology. In this approach, models were trained on data excluding a specific object and subsequently tested on the velocities associated with the omitted object. This process was repeated for each object, ensuring a comprehensive evaluation. The LSTM model architecture consists of three layers, each with 30 hidden units, a dropout rate of 0.2, and a sequence length of 5. The MLP model features a three-layer design, transitioning through 64 and 32 units, including Layer Normalization and a dropout rate of 0.1. Evaluation metrics, as defined earlier, were computed for each test case and averaged across all objects to assess the models' overall accuracy, as presented in Table 3.5. Both models demonstrated enhanced precision when trained with combined feature sets. Notably, the LSTM model exhibited significant improvements, reducing the Mean Absolute Error (MAE) to 0.26 cm/s, the Root Mean Square Error (RMSE) to 0.77 cm/s, and achieving an R^2 score of 0.75. Similarly, the MLP model showed performance gains, with the MAE decreasing to 0.30 cm/s, the RMSE reducing to 1.01 cm/s, and the R^2 score increasing to 0.68. Overall, the LSTM model consistently outperformed the MLP model, highlighting its superior capability in capturing and leveraging temporal data for slip severity estimation.

Generalization to unseen objects: Table 3.6 presents the performance comparison between LSTM and MLP models on unseen objects with identical slipping velocity profiles,

Table 3.5: Cross-validation results for Slip Severity Estimation

Metrics		MAE *(cm/s)*	RMSE *(cm/s)*	R^2
LSTM	Baseline	0.45	1.31	0.45
	Proposed	0.37	1.10	0.58
	Combined	**0.26**	**0.77**	**0.75**
MLP	Baseline	0.54	1.40	0.38
	Proposed	0.43	1.21	0.51
	Combined	**0.30**	**1.01**	**0.68**

evaluating their slip severity estimation accuracy. Metrics were calculated for each profile and averaged to assess overall effectiveness. The LSTM model consistently outperformed the MLP across all objects in terms of MAE and RMSE, highlighting its superior precision and reliability. Notably, the LSTM achieved an impressive R^2 value of 0.80 on the duster, demonstrating strong predictive capability. In comparison, the MLP's best R^2 was 0.66 on the same object, indicating moderate effectiveness. The LSTM also performed well on a challenging object like the deformable and porous sponge, where the MLP encountered significant difficulties.

Table 3.6: Results of trained Slip Severity Estimation models on unseen objects

Metrics		MAE *(cm/s)*	RMSE *(cm/s)*	R^2
 Book	**LSTM**	**0.65**	**0.92**	**0.65**
	MLP	0.98	1.51	0.54
 Scissors	**LSTM**	**0.53**	**0.90**	**0.67**
	MLP	0.89	1.11	0.58
 Plank	**LSTM**	**0.64**	**1.01**	**0.68**
	MLP	1.27	1.43	0.57
 Sponge	**LSTM**	**0.61**	**1.19**	**0.60**
	MLP	1.39	1.51	0.50
 Duster	**LSTM**	**0.42**	**0.68**	**0.80**
	MLP	0.67	0.93	0.66

Chapter 4

SLIP MITIGATION I: REAL-TIME GRIPPER CONTROL FOR SLIP-SENSITIVE OBJECT HANDLING

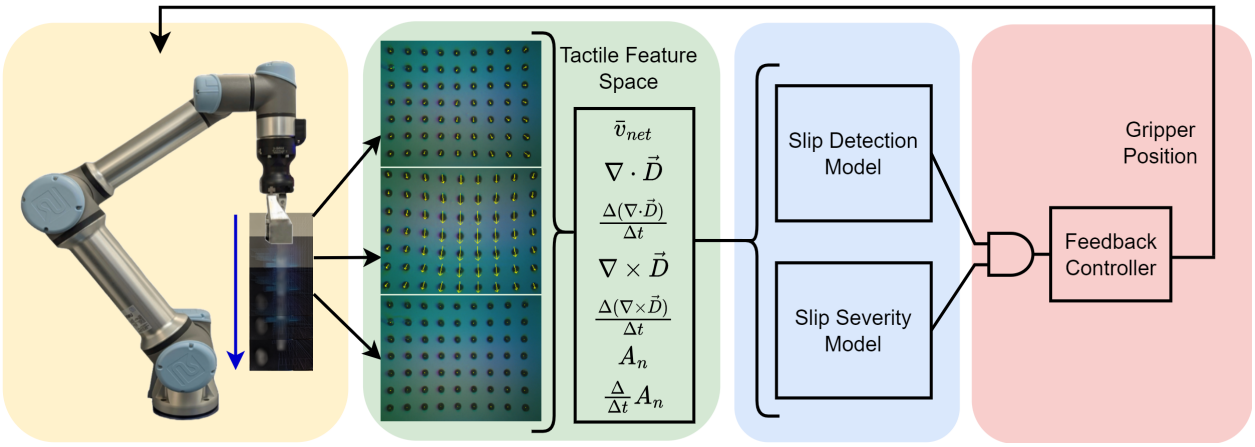

Figure 4.1: Summary of the Slip-Detection-Severity Framework: A robot executes an object handling task, during which tactile features from the GelSight Mini sensor are extracted in real time. These features simultaneously feed into the Slip Detection and Slip Severity models. Upon detecting slip, the feedback controller actively adjusts the gripper to mitigate slip severity.

4.1 Introduction

In the previous chapters, we developed slip detection and slip severity estimation models based on tactile vector field features. For slip-control tasks, these models must operate cohesively to ensure real-time performance and reliability. In industrial applications, integrating such models is vital for implementing slip mitigation algorithms, enhancing both safety and

efficiency in object handling.

To illustrate the practical utility of the proposed framework, this chapter demonstrates its application in a vertical sliding control task. The robot grasps objects with sufficient force to maintain stability while minimizing potential damage. As slip velocity increases, a proportional-derivative (PD) controller dynamically adjusts the grip force, mitigating slip in real time while avoiding excessive force. This experiment highlights the integration of slip detection and severity estimation in a realistic manipulation scenario, showcasing the effectiveness of the framework in achieving robust, adaptive control as shown in Fig 4.1

4.2 Task and Algorithm Details

With the development of the slip detection and slip severity models, we further implement a Proportional-Derivative (PD) controller on the Robotiq Hand-E gripper, which solely manipulates the gripper's position to adjust gripping force. The controller enhances grip by modulating the distance between the gripper fingers: the Robotiq Hand-E is fully open at a position value p of 0 and completely closed at 225. This control strategy, alongside the slip severity estimator, is activated upon slip detection to mitigate slip severity and is otherwise inactive to conserve computational resources and avoid unnecessary adjustments. Once slip velocity is ascertained by the estimation model, the gripper controller intervenes to enhance grip, thereby aiming to reduce the slip severity to zero. This approach demonstrates that effective control can be achieved using a straightforward position-controlled gripper within the proposed framework.

4.3 Experiment Details

This experiment integrates learned slip detection and severity estimation models into a cohesive framework for executing a vertical sliding task using a PD (Proportional-Derivative) controller, as described in 12. Leveraging their accuracy and generalizability to unseen, challenging objects, we use Random Forest for slip detection and LSTM for slip severity estimation.

Algorithm 1 PD Control for Slip Mitigation using HandE gripper

Inputs: S_{slip} : Slip detection signal

S_{severity} : Slip severity

p_{current} : Current gripper position

$S_{\text{target}} = 0$: Desired slip severity

K_p : Proportional gain

K_d : Derivative gain

Output: p_{new} : New gripper position

Initialize: $e_{\text{previous}} \leftarrow 0$

while Gripper is operational **do**
  if S_{slip} is detected **then**
    $e \leftarrow S_{\text{severity}} - S_{\text{target}}$
    $\dot{e} \leftarrow e - e_{\text{previous}}$
    $p_{\text{adjustment}} \leftarrow K_p \cdot e + K_d \cdot \dot{e}$
    $p_{\text{new}} \leftarrow \max(\min(p_{\text{current}} - p_{\text{adjustment}}, 225), 0)$
    Update gripper position to p_{new}
    $e_{\text{previous}} \leftarrow e$
  else
    Maintain current gripper position
  end if
end while

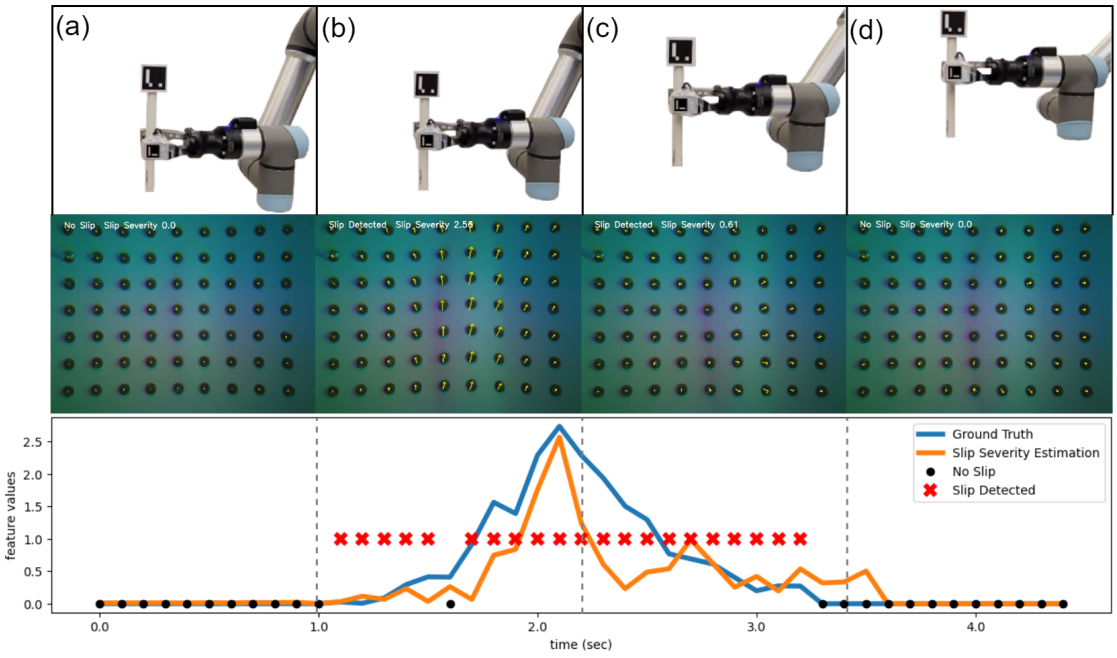

Figure 4.2: These four snapshots depict a tactile-guided vertical sliding control task. Panel (a) shows the robot grasping the PVC pipe and initiating vertical movement. In panel (b), acceleration leads to increased slip severity, which is concurrently detected and quantified. By panel (c), the robot adjusts its grip to effectively manage slip severity, avoiding overcorrection. Finally, in panel (d), the robot successfully mitigates the slip.

The demonstration object is a smooth PVC pipe, excluded from the training set. Its surface geometry, texture, and inertia are unfamiliar to the GelSight sensor. This object was chosen because its slippery surface poses significant challenges for tactile sensor-based slip detection, as observed in Li et al.'s work [35]. The robot operates at a target velocity of 3.8 cm/s. For ground truth comparison with slip severity estimation, one ArUco marker is affixed to the end effector and another to the top of the object. The markers' data is published on separate ROS topics at 60Hz and throttled to 25Hz through republishing with `throttle`. Furthermore, the ROS `ApproximateTime` policy synchronizes these topics, enabling concurrent pose data collection. An EWMA filter is applied to smooth the velocities,

mimicking the ground truth used for training the slip severity estimation model. After iterative tuning, the PD controller gains were set to $K_p = 3.10$ and $K_d = 0.42$.

Figure 4.2 illustrates the experiment's real-time operations. Panel (a) shows the robot initiating a grasp on the PVC pipe without detecting slip, where the slip severity estimate is correctly 0. The gripper's initial position is calibrated to the pipe's diameter plus a tolerance margin, inducing slight slip on the GelSight sensor surface. Once vertical motion begins, slip is detected, and the model provides real-time slip severity estimates. In panel (b), the slip severity model predicts a value of 2.56 cm/s from tactile data, closely matching the ground truth of 2.73 cm/s. The tuned PD controller adjusts the gripper's position based on the slip severity error, effectively mitigating slip. Notably, there is a single misclassification during the task, where the slip signal is momentarily set to 0.

In panel (c), the slip severity estimate decreases progressively, reflecting a reduction in relative velocity between the object and the sensor despite ongoing slip detection and severity estimation. The controller continues to reduce the error until the slip severity estimate reaches 0.00 cm/s, successfully stopping further slip. Even during the transition from (c) to (d), the slip detector's accuracy prevents overcorrection, even when the severity estimator predicts low values. The framework's performance on smooth objects like the PVC pipe, achieving an MAE of 0.21 cm/s, RMSE of 0.41 cm/s, and an R^2 of 0.68, demonstrates its effectiveness for time-dependent slip severity estimation.

Chapter 5

SLIP MITIGATION II: MINIMIZING SLIP THROUGH OPTIMIZED ROBOTIC TRAJECTORIES USING MODEL PREDICTIVE PATH INTEGRAL CONTROL

5.1 Introduction

In the preceding chapters, we developed a systematic framework for detecting slip and quantifying its severity. Additionally, we demonstrated the use of these learned models in a vertical sliding task, where a proportional-derivative (PD) controller was employed to regulate the gripper width and mitigate slip during object manipulation. However, certain scenarios present limitations, such as low-resolution grippers or jerky motion, where slip may be unavoidable due to hardware constraints. Addressing these challenges requires a shift in perspective, approaching the problem from the lens of trajectory optimization.

Humans are inherently adept at manipulation tasks like pick-and-place operations, not only because of their ability to grasp various objects appropriately but also because of their capacity to adjust motions based on object properties and placement goals in Cartesian space. Implicitly, humans optimize their motion based on the type of object being manipulated. Inspired by this, we address the problem of avoiding slip through trajectory optimization.

Trajectory optimization methods leverage predictive models of agents to simulate future trajectories and select the optimal current action [38]. Model Predictive Control (MPC) is one such widely-used method, wherein a locally optimal policy is computed over a predefined time horizon [22]. Starting from the current state and given a dynamics model, an action is selected from the computed policy, after which the optimization is repeated iteratively. This iterative process inherently reduces model bias by continuously updating the solution using the latest state information.

In the case of robotic manipulators, implementing MPC in joint space presents significant challenges. While gradient-based dynamics models can be used in joint space, they require cost functions to be differentiable and convex [47]. This constraint becomes problematic when dealing with non-convex or non-differentiable cost functions, which are often encountered in practical manipulation tasks. Recent studies suggest that sampling-based MPC methods excel in such scenarios. These methods can handle non-differentiable costs effectively and encode complex task constraints. Furthermore, sampling-based MPC approaches can be parallelized on GPUs, enabling faster and more robust implementations, as shown in [3]

In this chapter, we explore how sampling-based MPC, specifically Model Predictive Path Integral (MPPI) scheme implemented in the paper STORM [3], can be utilized to optimize robotic trajectories for minimizing slip. By formulating the problem within this framework, we aim to provide a robust approach for trajectory optimization in manipulator control that addresses the challenges posed by non-differentiable costs and hardware constraints. Concretely, this chapter makes the following contributions:

- We formulate the problem of slip minimization within the framework of trajectory optimization, focusing on the integration of slip as a control-relevant variable.

- We demonstrate a POC application of MPPI control to optimize robotic trajectories for minimal slip, using a simple analytical cost function.

5.2 Problem Setup

We address the problem of optimizing the trajectories of a manipulator with n joints tasked with picking an object and reaching a goal in task space, while avoiding slip. This is fundamentally a feedback control problem where the objective is to derive a policy that accounts for the non-differentiable constraints and costs associated with the task. Additionally, the policy must be robust to external disturbances that could induce slip during manipulation.

To frame this problem formally, we consider it as a Markov Decision Process (MDP). An MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$, where:

- $\mathcal{S}$ represents the state space, including the joint positions, velocities, and other relevant system variables.

- $\mathcal{A}$ represents the action space, corresponding to the commanded joint accelerations $\ddot{\theta}_{\text{commanded}}$.

- $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ defines the transition dynamics, capturing the system's evolution given the current state and action, potentially modeled using forward kinematics and dynamics.

- $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, or equivalently, the negative cost function, which encodes task objectives such as reaching the goal, maintaining stability, and minimizing slip.

- $\gamma \in [0, 1)$ is the discount factor, which balances immediate versus future rewards.

The objective of the optimization is to derive a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the expected cumulative reward:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right],$$

where the expectation is taken over the trajectory distribution induced by π.

In the context of trajectory optimization for manipulators, the optimization problem can be framed as minimizing a cost function J over a finite time horizon T, starting from an initial state s_0:

$$J = \sum_{t=0}^{T} \left[c_{\text{task}}(s_t, a_t) + c_{\text{constraints}}(s_t, a_t) + c_{\text{slip}}(s_t, a_t) \right].$$

Where:

- c_{task} represents task-specific costs, such as reaching a target position in task space or achieving a desired end-effector orientation.

- $c_{\text{constraints}}$ encodes constraints such as joint limits, collision avoidance, or system dynamics.

- c_{slip} penalizes slip-induced motion discrepancies, ensuring the object remains stable in the gripper.

The optimization must account for non-differentiable costs, such as collision penalties or slip-related constraints, which make traditional gradient-based methods unsuitable. Instead, sampling-based methods, such as MPPI, are employed. In this framework, the sampling process leverages the system's transition dynamics to simulate future states. The optimization iteratively selects actions that minimize the cost while accommodating non-differentiable task constraints. This approach inherently adapts to disturbances, as the policy updates dynamically based on real-time feedback from the system state.

This formulation builds upon recent work such as the STORM framework, which demonstrates the effectiveness of sampling-based methods for trajectory optimization in robotic manipulation, particularly in handling non-convex and non-differentiable constraints. By extending this approach to incorporate slip-aware cost, we aim to ensure robust manipulation that maintains stability and minimizes slip across a range of real-world scenarios.

5.3 Methodology

Similar to the STORM framework, we use sampling-based MPC to optimize open-loop control sequences represented as a sequence of time-independent Gaussian distributions. The policy π_θ is parameterized by the sequence of means $\mu_t = [\mu_{t,0}, \ldots, \mu_{t,H-1}]$ and covariances $\Sigma_t = [\Sigma_{t,0}, \ldots, \Sigma_{t,H-1}]$ over a finite horizon H.

At each iteration, the following steps are performed:

1. Sample N control sequences $\mathbf{u}_{k,h}$, where $k \in [0, N)$ and $h \in [0, H)$, from the current Gaussian policy.

2. Use an approximate dynamics model to propagate these control sequences and compute the resulting states $\hat{\mathbf{s}}_{k,h}$ and associated costs $\hat{\mathbf{c}}_{k,h}$.

3. Update the policy parameters μ_t and Σ_t using a sample-weighted gradient based on the objective function.

4. Execute the first action from the mean of the updated distribution or sample an action for execution.

The expected cost for a trajectory is defined as:

$$\hat{J}(\mathbf{s}_t, \mathbf{u}_t) = \sum_{h=0}^{H-2} \gamma^h \hat{c}(\hat{\mathbf{s}}_h, \mathbf{u}_h) + \gamma^{H-1} \hat{q}(\hat{\mathbf{s}}_{H-1}),$$

where $\gamma \in [0, 1]$ is a discount factor and $\hat{q}$ represents the terminal cost. The optimization objective is defined as:

$$L = \mathbb{E}_{\pi_\theta} \left[\exp\left(-\frac{1}{\beta} \hat{J}(\mathbf{s}_t, \mathbf{u}_t) \right) \right],$$

where β is a temperature parameter that controls exploration.

The policy parameters are updated as:

$$\mu_{t,h} = (1 - \alpha_\mu)\mu_{t-1,h} + \alpha_\mu \frac{\sum_{i=1}^{N} w_i \mathbf{u}_{i,h}}{\sum_{i=1}^{N} w_i},$$

$$\Sigma_{t,h} = (1 - \alpha_\Sigma)\Sigma_{t-1,h} + \alpha_\Sigma \frac{\sum_{i=1}^{N} w_i (\mathbf{u}_{i,h} - \mu_{t,h})(\mathbf{u}_{i,h} - \mu_{t,h})^\top}{\sum_{i=1}^{N} w_i},$$

where w_i represents the sample weights:

$$w_i = \exp \frac{-1}{\beta} \left(\sum_{h=0}^{H-2} \gamma^h \hat{c}(\hat{\mathbf{s}}_{h,i}, \mathbf{u}_{h,i}) + \gamma^{H-1} \hat{q}(\hat{\mathbf{s}}_{H-1,i}) \right).$$

5.3.1 Manipulator Kinematic Model

Leveraging the adaptability of Model Predictive Control (MPC) with simplified models, we utilize a kinematic transition model inspired by the STORM framework. The robot's joint-space state is defined as $\mathbf{s} = [\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}]$, where $\mathbf{q}$ represents joint positions, $\dot{\mathbf{q}}$ joint velocities, and

$\ddot{\mathbf{q}}$ joint accelerations. The control input is denoted as $\mathbf{a} = \ddot{\mathbf{q}}_{\text{cmd}}$, representing the commanded joint accelerations.

To efficiently propagate the system state over the control horizon H and for a batch of sampled control inputs, a parallelized computation is employed. Starting from an initial state $\mathbf{s}_0 = [\mathbf{q}_0, \dot{\mathbf{q}}_0, \ddot{\mathbf{q}}_0]$, the evolution of states for each sampled trajectory is computed as:

$$\ddot{\mathbf{Q}} = \mathbf{A}, \quad \dot{\mathbf{Q}} = \dot{\mathbf{q}}_0 + \Delta t \cdot \ddot{\mathbf{Q}}, \quad \mathbf{Q} = \mathbf{q}_0 + \Delta t \cdot \dot{\mathbf{Q}},$$

where $\mathbf{A}$ is the batch of sampled control inputs (joint accelerations) over the horizon, Δt is the time step, and $\mathbf{Q}$, $\dot{\mathbf{Q}}$, and $\ddot{\mathbf{Q}}$ are the joint positions, velocities, and accelerations over the horizon.

The resulting joint-space trajectories, $\hat{\mathbf{s}}$, are subsequently transformed into task-space representations, such as the end-effector pose, velocity, and acceleration, via forward kinematics and the Jacobian:

$$\mathbf{P} = \mathcal{FK}(\mathbf{Q}), \quad \dot{\mathbf{P}} = \mathcal{J}(\mathbf{Q})\dot{\mathbf{Q}}, \quad \ddot{\mathbf{P}} = \dot{\mathcal{J}}(\mathbf{Q})\dot{\mathbf{Q}} + \mathcal{J}(\mathbf{Q})\ddot{\mathbf{Q}},$$

where $\mathcal{FK}(\mathbf{Q})$ computes the forward kinematics, $\mathcal{J}(\mathbf{Q})$ is the Jacobian matrix, and $\dot{\mathcal{J}}(\mathbf{Q})$ is its time derivative.

This approach allows for the efficient computation of both joint-space and task-space states, enabling optimization of trajectories within the MPC framework.

5.3.2 Cost Terms

The optimization framework considers several key cost terms, which are summarized in Table 5.1. These costs ensure that the robot achieves the task objectives while adhering to physical and operational constraints.

Refer to the STORM framework [3] for more details on the formulation of reaching cost, joint limit cost, and self-collision avoidance cost. The formulation of slip cost is discussed in detail in the following sections.

Table 5.1: Cost Terms Used in Optimization

Cost Term	**Description**
Reaching Cost	Penalizes the Cartesian distance between the end-effector and the desired goal position and orientation.
Joint Limit Cost	Applies penalties when joint positions, velocities, or accelerations approach or exceed defined limits.
Self-Collision Avoidance Cost (jointNERF)	Penalizes configurations where parts of the robot are predicted to come into collision.
Slip Cost	Penalizes discrepancies between the predicted and desired velocities of the object relative to the gripper to minimize slip.

5.3.3 Analytical Slip Cost Formulation

To minimize slip during manipulation, we derive an analytical cost function that penalizes the relative velocity between the gripper and the object at contact points. This approach ensures that the object remains stable in the gripper while being manipulated. Below, we detail the formulation.

Object State: The object's state is represented as:

$$\mathbf{x}_{\text{object}} = [\mathbf{p}_{\text{object}}, \mathbf{v}_{\text{object}}],$$

where $\mathbf{p}_{\text{object}}$ is the object's position in Cartesian space, and $\mathbf{v}_{\text{object}}$ is the object's velocity.

Modeling Object Velocity: We model the object's velocity at each contact point i as a function of the gripper's velocity using the Jacobian:

$$\mathbf{v}_{\text{object},i} = \mathcal{J}_{\text{object},i}\dot{\mathbf{q}},$$

where $\mathcal{J}_{\text{object},i}$ is the Jacobian at the i-th contact point, mapping joint velocities $\dot{\mathbf{q}}$ to the object's velocity. The Jacobian can be decomposed as:

$$\mathcal{J}_{\text{object},i} = \begin{bmatrix} \mathcal{J}_{\text{linear}} \\ \mathcal{J}_{\text{angular}} \end{bmatrix} + \mathcal{J}_{\text{offset},i},$$

where $\mathcal{J}_{\text{linear}}$ and $\mathcal{J}_{\text{angular}}$ represent the contributions from linear and angular motion, and $J_{\text{offset},i}$ accounts for specific offsets at the contact point.

Weighted Object Velocity: To compute the object's overall velocity across n contact points, we take a weighted sum:

$$\mathbf{v}_{\text{object}} = \frac{\sum_{i=1}^{n} w_i \mathbf{v}_{\text{object},i}}{\sum_{i=1}^{n} w_i},$$

where w_i are the weights associated with each contact point. For simplicity, equal weights can be assumed ($w_i = 1$) but there are caveats to it as well.

Slip Cost: Slip occurs when there is a velocity mismatch between the object and the gripper. The relative velocity at each contact point is given by:

$$\mathbf{v}_{\text{relative},i} = \mathbf{v}_{\text{object},i} - \mathbf{v}_{\text{gripper},i},$$

where $\mathbf{v}_{\text{gripper},i}$ is the velocity of the gripper at the corresponding contact point.

The slip cost is then formulated as:

$$c_{\text{slip}} = \frac{1}{2}\|\mathbf{v}_{\text{object}} - \mathbf{v}_{\text{gripper}}\|^2,$$

Interpretation: This formulation penalizes any relative velocity between the gripper and the object, encouraging the manipulator to move in coordination with the object. The inclusion of weights w_i allows the model to prioritize certain contact points based on their significance. By minimizing this cost, the manipulator can effectively mitigate slip and improve stability during manipulation tasks.

5.4 Experiment and Discussion

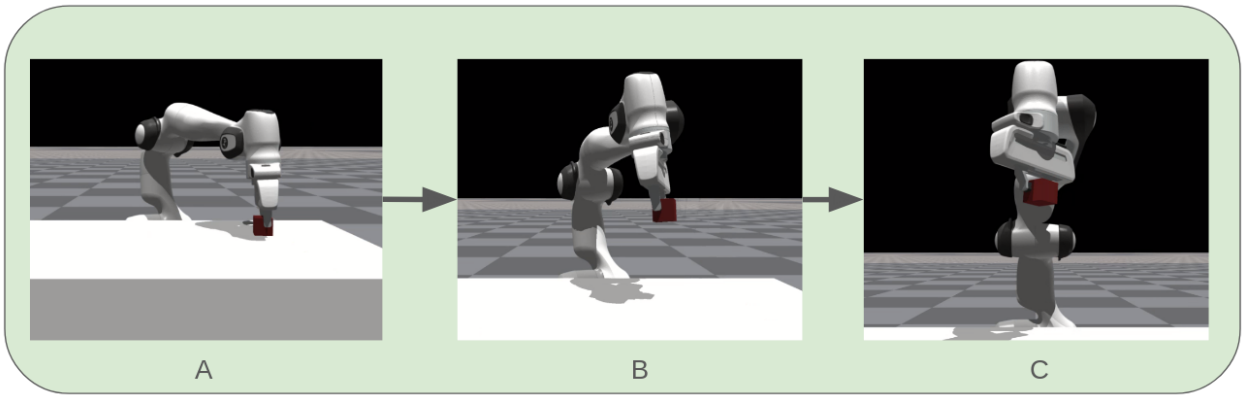

Figure 5.1: Demonstration of minimal slip trajectory optimization. (A) The robot grasps a cube placed on the table using a nominal grip width, defined as the object width plus a small tolerance. (B) The robot transports the cube using MPPI) control without considering minimal slip constraints. (C) The robot successfully reaches the Cartesian goal position while maintaining a secure grasp, preventing any slip of the cube.

We consider a scenario in which a Franka Panda manipulator is tasked with reaching a target pose in Cartesian space while carrying a cube in its gripper. The objective is to follow a trajectory that minimizes slip during the manipulation task. To evaluate the performance, we compare the one-time pick success rates for MPPI with and without incorporating an analytical slip cost.

Figure 5.1 illustrates the improvement in success rates achieved by including a simple

analytical kinematic slip cost alongside a simplified forward model for the object. The success rate for the pick task increases from 10% to 70% when the slip cost is included. This result demonstrates the potential of MPC as an effective methodology for optimizing trajectories to minimize slip, providing a POC for further exploration of slip-aware control strategies. A promising direction for future work involves integrating dynamics-based friction costs to enable more formal and robust optimization. Additionally, incorporating perception within the control loop could facilitate the development of hierarchical, task-specific policies aimed at minimizing slip, thereby enhancing the generalization of manipulation strategies.

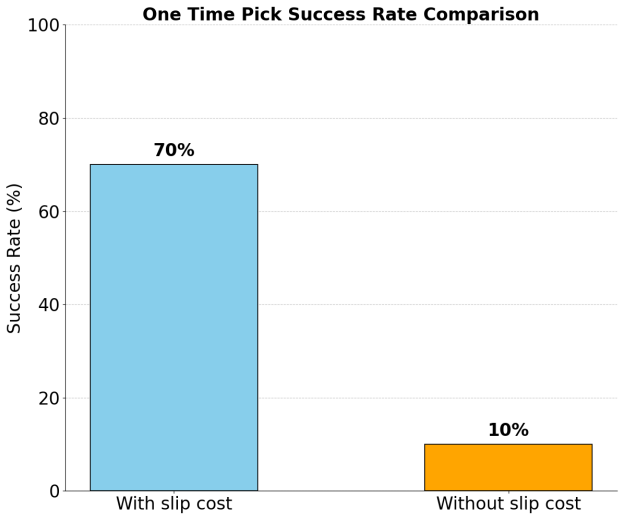

Figure 5.2: Success rates for cartesian transport of the cube, evaluated over 10 episodes with randomly sampled start and goal locations within the manipulator's workspace.

Chapter 6
CONCLUSION AND FUTURE WORK

This thesis concludes with a summary of its primary contributions and key insights, along with a discussion of potential future directions for the research presented.

6.1 Conclusive Contributions and Key Takeaways

6.1.1 Conclusive Contributions

The primary contributions of this thesis are as follows:

- A method was developed to measure deformation in the tactile sensor gel using Optical Flow, enabling the extraction of features relevant to surface interactions.

- Extensive tactile feature engineering was conducted to identify relationships between vector flow-based tactile fields and various surface interaction forces. This effort facilitated the learning of tactile contact representations for the phenomenon of slip.

- A Learned Slip Detection Severity framework was proposed by training machine learning models to detect slip and quantify its severity. A hierarchical approach for slip-aware manipulation was demonstrated, showcasing the utility of the framework even when using a simple position-controlled gripper. The methodology and results were documented in the publication, "Learned Slip-Detection-Severity Framework using Tactile Deformation Field Feedback for Robotic Manipulation" [29].

- Preliminary work was presented on using sampling-based optimization methods to compute minimal slip paths, leveraging a simple analytical model and cost function for object slip.

6.1.2 Key Takeaways

The key insights derived from this thesis are as follows:

- Slip is not merely a binary phenomenon; it can be quantified and optimized within robotic control frameworks, provided reliable methods for slip measurement and representation are established.

- The problem of slip in robotic manipulation can be addressed from two complementary perspectives - gripper based control and trajectory optimization. Combining these approaches can enable the development of more advanced and adaptive slip-aware robotic systems.

- In certain scenarios, simpler models and classifiers, when combined with effective feature extraction techniques, can achieve comparable performance to neural networks, particularly when data availability is limited.

6.2 Future Work

The research presented in this thesis opens several avenues for future exploration:

- Incorporating advanced gripper designs with force and position control capabilities to enhance the applicability and robustness of the proposed methods.

- Exploring reinforcement learning-based approaches to refine slip-aware manipulation strategies, integrating the slip severity model as feedback for policy optimization, using newly developed tactile simulators like TacSL [2].

- Investigating the frictional dynamics formulation to encode slipping as a robust cost in optimization methods as opposed to simple approximate analytical costs.

- Developing generalized benchmarks for slip-aware manipulation to evaluate and compare the performance of different tactile sensing and control methodologies.

BIBLIOGRAPHY

[1] Ioannis Agriomallos, Stefanos Doltsinis, Ioanna Mitsioni, and Zoe Doulgeri. Slippage detection generalizing to grasping of unknown objects using machine learning with novel features. *IEEE Robotics and Automation Letters*, 3(2):942–948, 2018.

[2] Iretiayo Akinola, Jie Xu, Jan Carius, Dieter Fox, and Yashraj Narang. Tacsl: A library for visuotactile sensor simulation and learning, 2024.

[3] Mohak Bhardwaj, Balakumar Sundaralingam, Arsalan Mousavian, Nathan D. Ratliff, Dieter Fox, Fabio Ramos, and Byron Boots. Storm: An integrated framework for fast joint-space model-predictive control for reactive manipulation. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 750–759. PMLR, 08–11 Nov 2022.

[4] A. Bicchi and V. Kumar. Robotic grasping and contact: a review. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 1, pages 348–353 vol.1, 2000.

[5] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics*, 30(2):289–309, 2014.

[6] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[7] Guanqun Cao, Jiaqi Jiang, Danushka Bollegala, Min Li, and Shan Luo. Multimodal zero-shot learning for tactile texture recognition, 2023.

[8] Guanqun Cao, Jiaqi Jiang, Chen Lu, Daniel Fernandes Gomes, and Shan Luo. Touchroller: A rolling optical tactile sensor for rapid assessment of large surfaces. 2021.

[9] Yuan Chen, Colin Prepscius, Daewon Lee, and Daniel D. Lee. Tactile velocity estimation for controlled in-grasp sliding. *IEEE Robotics and Automation Letters*, 6(2):1614–1621, 2021.

[10] Marco Costanzo, Giuseppe De Maria, and Ciro Natale. Detecting and controlling slip through estimation and control of the sliding velocity. *Applied Sciences*, 13(2), 2023.

[11] Luke Cramphorn, John Lloyd, and Nathan F. Lepora. Voronoi features for tactile sensing: Direct inference of pressure, shear, and contact locations, 2018.

[12] Mehmet Dogar, Kaijen Hsiao, Matei Ciocarlie, and Siddhartha Srinivasa. Physics-based grasp planning through clutter. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[13] Mehmet R. Dogar and Siddhartha S. Srinivasa. Push-grasping with dexterous hands: Mechanics and a method. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2123–2130, 2010.

[14] Siyuan Dong, Wenzhen Yuan, and Edward H. Adelson. Improved gelsight tactile sensor for measuring geometry and slip. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 137–144, 2017.

[15] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[16] Scott Funkhouser. The entropy of a discrete real variable. *Entropy*, 14(8):1522–1538, 2012.

[17] Kanishka Ganguly, Pavan Mantripragada, Chethan M. Parameshwara, Cornelia Fermüller, Nitin J. Sanket, and Yiannis Aloimonos. Gradtac: Spatio-temporal gradient based tactile sensing, 2022.

[18] Bin Gao, Shuai Yang, Haiyang Jin, Ying Hu, Xiaojun Yang, and Jianwei Zhang. Design and analysis of underactuated robotic gripper with adaptive fingers for objects grasping tasks. In *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 987–992, 2016.

[19] Yuri Gloumakov and Tae Myung Huh. Learning slip with a patterned capacitive tactile sensor. In *ICRA 2022 Workshop: Reinforcement Learning for Contact-Rich Manipulation*, 2022.

[20] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[21] Liang He, Qiujie Lu, Sara-Adela Abad, Nicolas Rojas, and Thrishantha Nanayakkara. Soft fingertips with tactile sensing and active deformation for robust grasping of delicate objects. *IEEE Robotics and Automation Letters*, 5(2):2714–2721, 2020.

[22] Lukas Hewing, Kim P. Wabersich, Marcel Menner, and Melanie N. Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(Volume 3, 2020):269–296, 2020.

[23] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997.

[24] E.G.M. Holweg, H. Hoeve, W. Jongkind, L. Marconi, C. Melchiorri, and C. Bonivento. Slip detection by tactile sensors: algorithms and experimental results. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3234–3239 vol.4, 1996.

[25] Xiaohai Hu, Aparajit Venkatesh, Guiliang Zheng, and Xu Chen. Learning to detect slip through tactile measures of the contact force field and its entropy, 2023.

[26] Zhixian Hu, Lan Lin, Waner Lin, Yingtian Xu, Xuan Xia, Zhengchun Peng, Zhenglong Sun, and Ziya Wang. Machine learning for tactile perception: Advancements, challenges, and opportunities. *Advanced Intelligent Systems*, 5(7):2200371, 2023.

[27] A. Ikeda, Y. Kurita, J. Ueda, Y. Matsumoto, and T. Ogasawara. Grip force control for an elastic finger using vision-based incipient slip feedback. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 1, pages 810–815 vol.1, 2004.

[28] Jasper Wollaston James and Nathan F. Lepora. Slip detection for grasp stabilization with a multifingered tactile robot hand. *IEEE Transactions on Robotics*, 37(2):506–519, 2021.

[29] Neel Jawale, Navneet Kaur, Amy Santoso, Xiaohai Hu, and Xu Chen. Learned slip-detection-severity framework using tactile deformation field feedback for robotic manipulation, 2024.

[30] Mohsen Kaboli, Philipp Mittendorfer, Vincent Hügel, and Gordon Cheng. Humanoids learn object properties from robust tactile feature descriptors via multi-modal artificial skin. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 187–192, 2014.

[31] Sing Bing Kang and K. Ikeuchi. Toward automatic robot instruction from perception-recognizing a grasp from observation. *IEEE Transactions on Robotics and Automation*, 9(4):432–443, 1993.

[32] Hayato Kanno, Hiroyuki Nakamoto, Futoshi Kobayashi, Fumio Kojima, and Wataru Fukui. Slip detection using robot fingertip with 6-axis force/torque sensor. In *2013 IEEE Workshop on Robotic Intelligence in Informationally Structured Space (RiiSS)*, pages 1–6, 2013.

[33] Gen-Ichiro Kinoshita, Shuhei Aida, and Masahiro Mori. A pattern classification by dynamic tactile sense information processing. *Pattern Recognition*, 7(4):243–251, 1975.

[34] Matthew Levins and Haoxiang Lang. A tactile sensor for an anthropomorphic robotic fingertip based on pressure sensing and machine learning. *IEEE Sensors Journal*, 20:13284–13290, 2020.

[35] Jianhua Li, Siyuan Dong, and Edward H. Adelson. Slip detection with combined tactile and visual information. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7772–7777, 2018.

[36] Tong Li, Yuhang Yan, Chengshun Yu, Jing An, Yifan Wang, and Gang Chen. A comprehensive review of robot intelligent grasping based on tactile perception. *Robotics and Computer-Integrated Manufacturing*, 90:102792, 2024.

[37] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics, 2017.

[38] Danylo Malyuta, Yue Yu, Purnanand Elango, and Behçet Açıkmeşe. Advances in trajectory optimization for space vehicle control. *Annual Reviews in Control*, 52:282–315, 2021.

[39] Hamza Merzic, Miroslav Bogdanovic, Daniel Kappler, Ludovic Righetti, and Jeannette Bohg. Leveraging contact forces for learning to grasp, 2018.

[40] Tingting Mi, Dashun Que, Senlin Fang, Zhenning Zhou, Chaoxiang Ye, Chengliang Liu, Zhengkun Yi, and Xinyu Wu. Tactile grasp stability classification based on graph convolutional networks. In *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 875–880, 2021.

[41] Andrés Montaño and Raúl Suárez. Manipulation of unknown objects to improve the grasp quality using tactile information. *Sensors*, 18(5), 2018.

[42] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation, 2019.

[43] Joaquín Royo Miquel, Masashi Hamaya, Cristian Beltran, and Kazutoshi Tanaka. Learning robotic assembly by leveraging physical softness and tactile sensing. 10 2023.

[44] B. Siciliano and O. Khatib. Sensing and perception. In *Springer Handbook of Robotics*, pages 455–471, 2008.

[45] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds, 2017.

[46] Benjamin Ward-Cherrier, Nicholas Pestell, Luke Cramphorn, Benjamin Winstone, Maria Elena Giannaccini, Jonathan Rossiter, and Nathan F. Lepora. The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies. *Soft Robotics*, 5(2):216–227, 2018.

[47] Zhaoming Xie, C. Karen Liu, and Kris Hauser. Differential dynamic programming with nonlinear constraints. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 695–702, 2017.

[48] Wenzhen Yuan, Siyuan Dong, and Edward H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12), 2017.

[49] Wenzhen Yuan, Rui Li, Mandayam A. Srinivasan, and Edward H. Adelson. Measurement of shear and slip with a gelsight tactile sensor. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 304–311, 2015.

[50] Yazhan Zhang, Zicheng Kan, Yang Yang, Yu Alexander Tse, and Michael Yu Wang. Effective estimation of contact force and torque for vision-based tactile sensors with helmholtz–hodge decomposition. *IEEE Robotics and Automation Letters*, 4(4):4094–4101, 2019.