Slip-Aware Robotic Handling: Tactile Object Characterization for Safe Manipulation

Navneet Kaur

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

Master of Science in Mechanical Engineering

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 Handling:
Tactile Object Characterization
for Safe Manipulation

Navneet Kaur

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

Slip is a significant challenge in manipulative tasks involving object gripping. Most research has focused on hard or deformable objects rather than simultaneously addressing both. This study develops and implements a robust slip detection mechanism that effectively identifies slips across various types of objects. The study also introduces a method for classifying objects based on tactile sensor images to determine whether they are hard or deformable. A gripper control method that utilizes this classification to enhance the handling of different object types is proposed.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

UR: Universal Robots

VIT: Vision Transformer

SPT: Shifted Patch Tokenization

LSA: Locality Self-Attention

ACKNOWLEDGMENTS

I wish to express my gratitude to my research advisor and Master's thesis committee chair, Professor Xu Chen, for his invaluable guidance throughout this project. I also extend my thanks to my thesis committee members, Professor Santosh Devasia and Professor Steve Brunton, for their mentorship and support. Additionally, I am grateful to the members of the MACS Lab at the University of Washington, particularly Neel Jawale, Xiaohai Hu, and Amy Santoso, for their assistance throughout my research.

DEDICATION

to my parents and friends and their unwavering support and encouragement

Chapter 1

INTRODUCTION

1.1 Motivation

Humans possess the ability to hold and feel objects in their hands, allowing them to sense when an object is slipping and to adjust their grip accordingly [11]. This capability enables humans to prevent slipping while avoiding damage to the object, regardless of its type. Unfortunately, robots lack this innate sense. Tactile sensors serve as a crucial tool to bridge this gap, providing robots with a means to approximate the human ability to sense and adjust to the objects they hold [8].

Gripper control is essential in robotic manipulation because objects vary in size and material properties; they can be either hard or deformable. Designing a gripper control scheme that can handle multiple types of objects is crucial. Hard objects create a distinct outline on tactile sensors, making them easier to detect and grasp accurately. However, soft and deformable objects exert slight pressure on the sensors, leading to challenges. In these cases, there is a risk of applying too much force, potentially causing permanent damage to the object.

On the other hand, applying too little force can result in the object slipping and losing contact with the gripper. Therefore, there is a need for a gripper control scheme that not only detects the occurrence of slip but also considers the type of object being manipulated. This allows the gripper to adjust its positions and force application accordingly, maintaining the object's integrity while preventing slip.

Alternative methods for handling objects and managing slips during manipulative tasks exist, such as using perception-based techniques to determine the position and orientation of the object [20, 15, 6]. However, these solutions have limitations. Perception can be

unreliable in poor lighting conditions or when occlusions block the camera's view. Even under ideal lighting conditions, perception-based solutions cannot provide information about the texture and material of the objects. Tactile sensors offer a solution that overcomes these challenges. In close contact with the objects, tactile sensors provide direct feedback on the object's properties [1]. This allows for the extraction of detailed information, enabling the categorization of objects based on their physical characteristics. Integrating tactile sensing into the manipulation process makes achieving more reliable and versatile control possible, particularly in scenarios where perception alone is insufficient.

1.2 Literature Review

1.2.1 Slip Detection and Estimation

Effective velocity control in robotic operations is critical for optimizing stowing and pullout tasks, where achieving a high loop rate is essential for efficiency and damage prevention. Key challenges include managing clutter within bins using advanced perception techniques, minimizing in-bin amnesty, and enhancing the robot's speed and precision.

Recent studies, such as those by Li et al. [17], address clutter management through transformer models applied to synthetic datasets for object segmentation and tracking. Nonetheless, issues related to unnecessary contact with objects and bins persist, especially in dynamic tasks where slip-and-slide events can lead to manipulation failures. Traditional tactile feedback-based grip controllers, as discussed in [10], attempt to mitigate slip by increasing normal force; however, this method often proves inadequate for handling delicate objects [12].

Understanding slip requires extracting relevant information from tactile sensors. Slip detection often involves analyzing Coulomb friction effects to account for both translational and rotational motions [18]. Techniques such as monitoring changes in force and frequency over time during slip events are also employed [7, 19]. Additionally, some approaches focus on detecting incipient slip by leveraging the frictional properties of elastomeric gel surfaces

in combination with forces and torques measured by tactile sensors [2, 25].

In this thesis, a slip detection model has been developed to identify slip occurrences for objects with diverse surface properties and deformability. This model demonstrates generalizability with an average accuracy of 92% on previously unseen objects.

1.2.2 Vision Transformers

Tactile-based stable grasping tasks have focused on enhancing grasp stability by detecting rotational displacement and predicting optimal grasping locations. Techniques employed include both model-based algorithms and model-free Reinforcement Learning (RL) methods [16, 13]. While these approaches effectively improve grasp stability in many scenarios, they generally do not address objects with varied surface properties that change over time. Consequently, these methods may not fully optimize gripping force for objects whose physical characteristics vary during manipulation.

Additionally, these methods often assume a constant weight distribution for objects throughout an episode. This assumption limits their applicability to scenarios where objects have changing weight distributions. As a result, the effectiveness of these grasping techniques may be reduced when dealing with objects that experience significant changes in weight or distribution during handling. Addressing these limitations requires advancements in grasping strategies that can adapt to dynamic object properties and varying weight distributions.

1.3 Experimental Setup

The experimental setup, as seen in Figure 1.1, includes a Robotiq Hand-E [22] flange attached to a UR5e [23] robotic arm with a maximum reach of 850mm. This cobot is mounted on a Vention table and is controlled using a teach pendant. The gripper fingers were custom-made in the lab to incorporate GelSight Mini tactile sensors [5]. These sensors are gel-based and provide real-time 2D or 3D images of the surface. They accurately map surface interactions

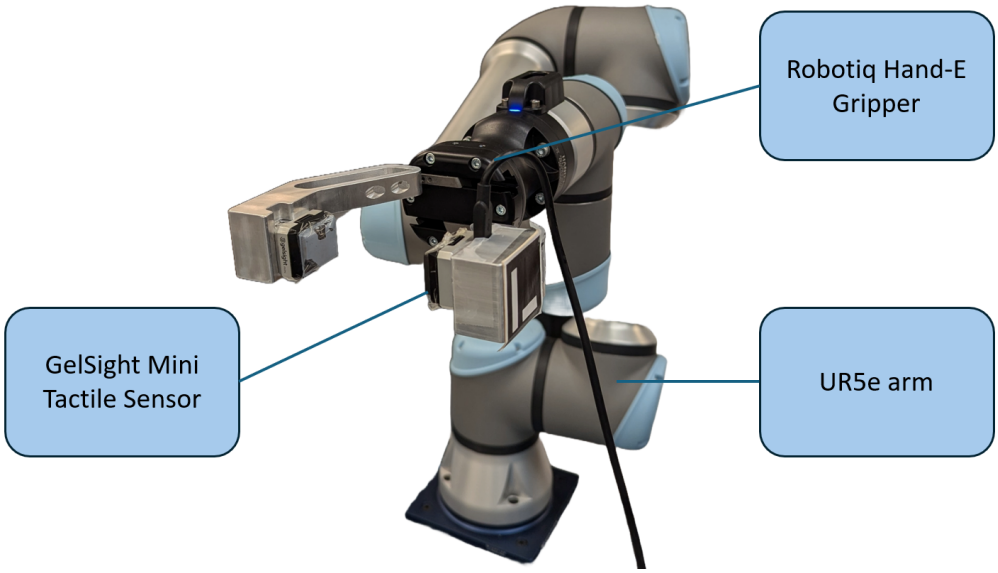


Figure 1.1: Experimental setup for the slip detection and object characterization model, including the tactile sensors

by capturing the details of objects pressed against the gel surface. The deformable elastomer-based surface of the sensors can capture high-resolution geometric features.

Although the setup includes two tactile sensors, only one was used for data collection and testing throughout the project. The tactile sensor is connected to a computing source via a USB-B cable. The computing system runs Ubuntu 20.04 and has an NVIDIA GeForce RTX 3050 GPU.

Chapter 2

SLIP DETECTION AND ESTIMATION

The GelSight Mini [5] is an elastomeric sensor that provides images of surface topography regardless of lighting conditions. It is a gel-based tactile sensor with a built-in camera that delivers live 2D images of the gel's surface when connected to a device. While these images reveal how the object interacts with the gel surface, they do not fully capture the nuances needed for effective analysis. To address this, extracting more relevant and interpretable features from the sensor data tailored to the specific problem at hand is essential. This feature extraction process enhances the utility of the tactile information provided by the GelSight Mini sensor. [1]

2.1 Feature Extraction

The gel surface is embedded with 63 equidistant markers arranged in a 7x9 grid. These markers appear as black dots on the tactile frame images, as shown in Figure 2.1. Each marker is tracked using the SimpleBlobDetector function in the OpenCV module, and its position is recorded to determine its displacement from the initial frame at rest [4]. This data is then used to calculate the displacement of each marker relative to its initial position, providing the deformation field of the entire surface. In addition to the net deformation, the following features are extracted:

1. **Velocity**: The rate of change of displacement for each marker is recorded as the velocity of the deformation field. The deformation field changes when the object in contact with the gel surface moves, such as during a slip between the object and the

[1] In collaboration with Neel Jawale

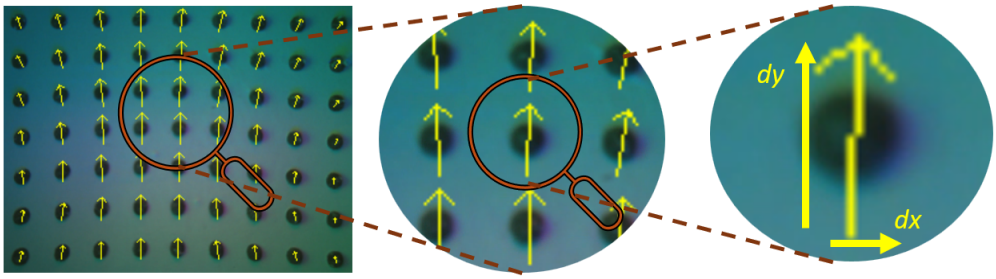


Figure 2.1: Tactile frame image and marker displacement calculation

sensor. The rate at which this deformation field changes is a critical feature for analysis.

$$\bar{v_k} = \frac{1}{63} \sum_{i=1}^{63} \frac{\Delta d_k}{\Delta t} \tag{2.1}$$

Here, v_k is the net velocity, and d_k is the displacement in k-direction, calculated over a Δt time interval.

2. **Divergence**: The divergence of the deformation field becomes notable in scenarios where changes in the applied normal force occur. This metric captures the expansion or contraction of the surface caused by the object being grasped.

$$\Delta \cdot \vec{d} = \frac{\Delta d_x}{\Delta x} + \frac{\Delta d_y}{\Delta y} \tag{2.2}$$

The divergence is recorded for each marker, and the net divergence is calculated for all 63 markers.

3. **Curl**: Curl of the deformation field captures the effect of rotational forces acting between the two surfaces.

$$\Delta \times \vec{d} = \frac{\Delta d_x}{\Delta x} - \frac{\Delta d_y}{\Delta y} \tag{2.3}$$

The curl for each marker is recorded, and the net curl is calculated for the 63 markers.

4. **Area in contact**: During manipulation tasks, monitoring any changes in the percentage of the area in contact between the object and the gel surface is crucial. This helps

in understanding the dynamics of the task and the interactions between the object and the sensor.

2.2 Slip Detection

2.2.1 Motivation

A vital aspect of this research was accurately detecting when an object was slipping against the sensor surface. To achieve this, two machine learning classifiers, the Random Forest (RF) classifier and the Gradient Boost (GB) classifier, were employed to classify whether a slip was occurring in real time.

2.2.2 Methodology

Fifteen different objects were chosen to train the machine learning models. These objects are shown in Figure 2.2. Three scenarios were carefully selected to ensure the data was accurately collected:

1. **Grasp Scenario**: Each object was placed between the gripper fingers, and the gripper position was incrementally adjusted for 1tencounts.

2. **No Slip Scenario**: Each object was securely grasped to prevent any motion that could induce slip. The robot then followed various trajectories while holding the object for 5 minutes, ensuring no slip occurred.

3. **Slip Scenario**: The object was manually induced to slip against the gel surface of the sensor for 5 minutes, ensuring that all collected data points corresponded to slip occurrences.

The grasp and no-slip scenarios were marked as no-slip occurrences and labeled as 0, while the slip scenario was specifically labeled as 1. After conducting all these experiments,

Figure 2.2: Objects used for training slip detection model. Objects were chosen for their varied texture and shape to make the slip detection model more generalized.

the data collected was meticulously pre-processed to remove any noise. Additionally, all annotations were carried out manually to ensure precision.

2.2.3 Results

The models are evaluated based on stratified 5-fold cross-validation. The results are surmised in Figure 2.3. The models were compared based on four metrics - Accuracy, F1 Score, Precision, and Recall. It is noted that the RF model performs better than the GB model. Hence, throughout the project, the RF model is chosen as a method of detecting the occurrence of slip.

2.3 Slip Severity Estimation

2.3.1 Motivation

While the classifiers effectively detected the occurrence of slip, they did not indicate the extent of the slip. It is critical to assess slip severity because the gripper control must respond before the object entirely slips out and falls. To address this, a slip severity estimation was defined using the same features as those used for the classifiers.

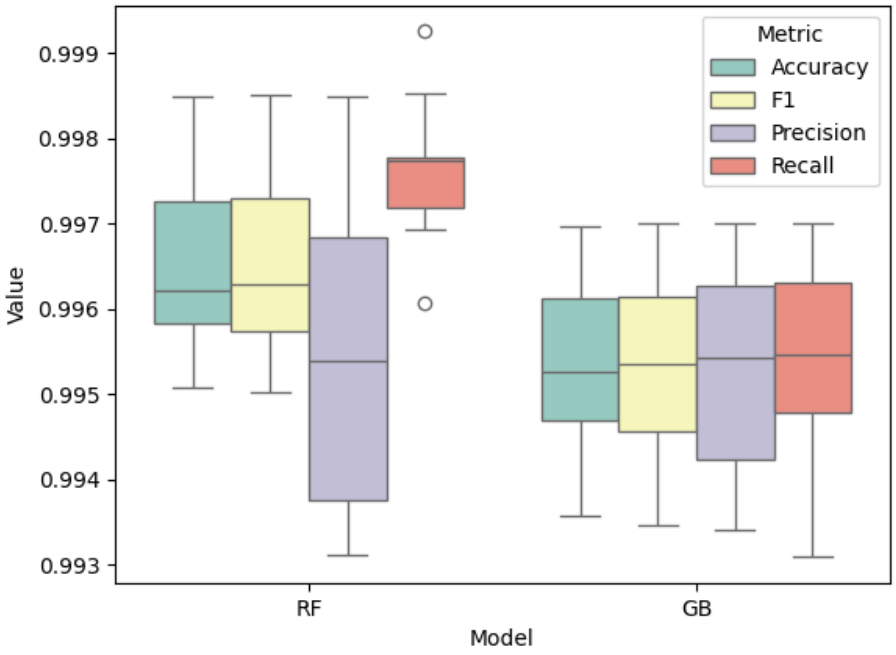


Figure 2.3: Comparison between RF and GB

2.3.2 Methodology

Two types of neural network models were employed to build this slip severity estimation: Long Short-Term Memory (LSTM) and Multi-Layer Perceptron (MLP). The data collection for this experiment was slightly different. Each object was securely placed between the gripper fingers, ensuring it remained stationary while the robot moved around it. The sensor maintained continuous contact with the object throughout the process. The robot executed a vertical motion, capturing the velocity at which the gripper moved against the object—this velocity represented the slipping velocity. This experiment was conducted at ten different velocities for each object. The train objects chosen for this experiment were the same as those used in slip detection model training and can be found in Figure 2.2.

Chapter 3

GRIPPER POSITION CONTROL

This section aims to define a model for tactile sensor features and to determine the inverse inputs required for the gripper to achieve the desired trajectory of the slip estimation. All experiments in this section are conducted using a metal rod, specifically a cuboid metal rod. This object was selected to study the system dynamics due to its smooth outer surface and sufficient length, which provides ample time to collect extensive data.

The PySINDy module [3] is utilized to derive the system equations for this project. This module requires system parameters as inputs for functions of time. It returns the derivatives of the given variables as functions of the input and parameters. Specifically, if $x(t)$ represents the chosen system parameters and u is the input to the system, the output will be an array of equations in the following form:

$$\dot{x} = f(x, u) \tag{3.1}$$

With the help of this module, the system dynamics are then calculated for gripper control.

3.1 Data Collection and Important Parameters

The sensor's gel surface is marked with small black markers to track the gel's displacement. The following parameters are used to analyze the sensor data:

1. Slip estimation (s): Measures the speed and extent of the object's slip against the gel surface of the tactile sensor.

2. Divergence of the Displacement Vector Field (d): Indicates how the displacement field is spreading out.

3. Rate of Change of Divergence (dd): Represents the rate at which the divergence of the displacement vector field is changing.

4. Curl of the Displacement Vector Field (c): Measures the rotation or twisting of the displacement vector field.

5. Rate of Change of Curl (dc): This indicates how the curl of the displacement vector field varies over time.

6. Percent of the Gel Area in Contact with the Object (a): Represents the proportion of the gel surface that is in contact with the object.

7. Rate of Change of Contact Area (da): Measures how the area of contact between the gel and the object is changing.

To determine the system dynamics, the following three experiments were conducted, with system parameters collected in real-time:

1. Grasping the object with varying gripper finger positions: The object was grasped while varying the position of the gripper fingers.

2. Grasping the object with incremental changes in gripper finger position: The object was grasped while the position of the gripper fingers changed incrementally.

3. Repeating experiment 2 with vertical arm movement: The procedure in experiment 2 was repeated while the robotic arm moved in a straight vertical line, causing the object to slip against the sensors.

3.2 System Dynamics

After the data for the three experiments is recorded, it is pre-processed to remove any noise. The data is then used to determine the relation for the time derivatives of the state variables.

The state variables are set as follows,

$$X = \begin{bmatrix} slip\ estimation \\ divergence \\ rate\ of\ change\ of\ divergence \\ curl \\ rate\ of\ change\ of\ curl \\ percent\ of\ area\ in\ contact \\ rate\ of\ change\ of\ area\ in\ contact \end{bmatrix} = \begin{bmatrix} s \\ d \\ dd \\ c \\ dc \\ a \\ da \end{bmatrix} \tag{3.2}$$

The change in position of the gripper finger is recorded as the input to the system and stored as u. The time derivative of X is determined as follows with the help of PySINDy,

$$\begin{aligned}
\dot{s} &= -0.454s - 1.897d - 0.236dd + 7.477c + 0.825dc + 2.793a + 0.492da \\
&\quad + 0.147s^2 + 1.154d^2 - 31.430c^2 + 1.582dc^2 - 2.634a^2 \\
\dot{d} &= dd \\
\dot{dd} &= 10.199s + 17.524d + 0.549dd - 53.334c + 0.391dc - 23.795a + 0.561da \\
&\quad - 0.257u - 2.301s^2 - 10.863d^2 + 191.650c^2 + 0.356dc^2 + 7.019a^2 \\
\dot{c} &= dc \\
\dot{dc} &= 0.944s + 2.451d - 11.155c + 0.735dc - 2.424a - 0.224s^2 - 1.311d^2 \\
&\quad + 24.932c^2 + 0.186dc^2 + 0.795a^2 \\
\dot{a} &= da \\
\dot{da} &= 9.376s + 17.169d + 0.164dd - 49.205c - 0.056dc - 19.844a + 0.925da \\
&\quad - 0.456u - 2.302s^2 - 9.271d^2 + 202.462c^2 - 0.383dc^2
\end{aligned} \tag{3.3}$$

The dynamics model is then tested on sample test data, and the predicted and the actual values of slip estimation are plotted to compare the model's accuracy. This plot can be seen in Figure 3.1. It can be noted that the predicted values (shown in blue) are noisy. This

can be attributed to the fact that for this project's scope, the relations between the state variables and their time derivatives were restricted to quadratic.

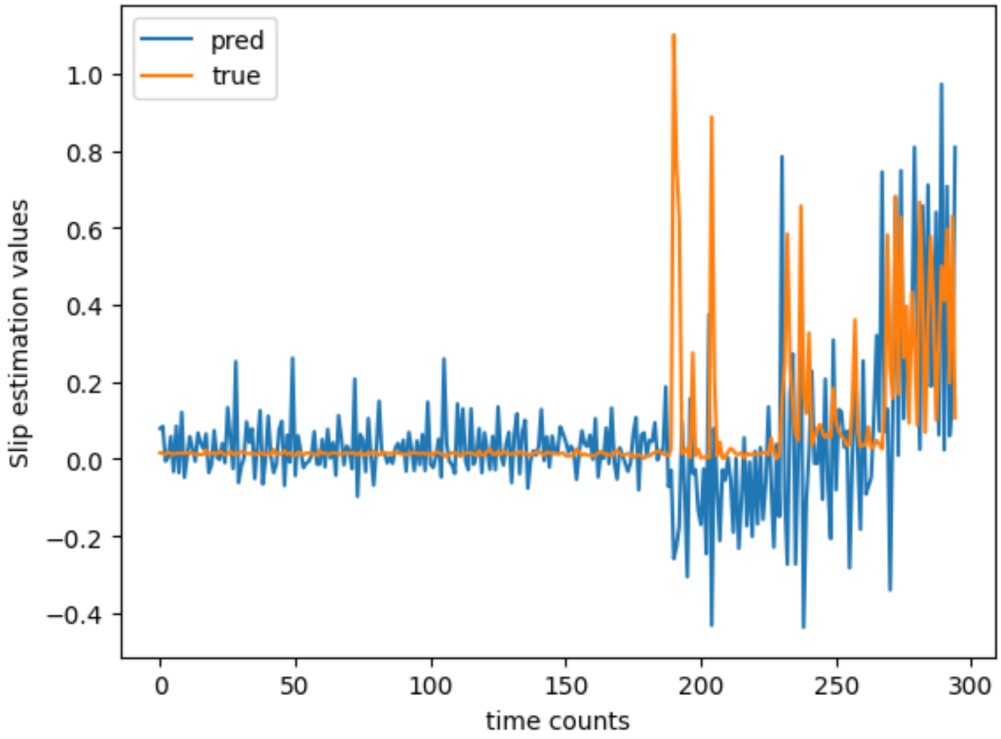


Figure 3.1: Comparison between the actual (labeled as true) values and predicted (labeled as pred) values of the slip estimation.

Therefore, the system dynamics can be written as,

$$\dot{X} = \begin{bmatrix} -0.4540 & -1.8970 & -0.2360 & 7.4770 & 0.8250 & 2.7930 & 0.4520 \\ 0 & 0 & 1.0000 & 0 & 0 & 0 & 0 \\ 10.1830 & 17.5240 & 0.5490 & -53.3340 & 0.3910 & -23.7950 & 0.5610 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 & 0 \\ 0.9440 & 2.4510 & 0 & -11.1550 & 0.7350 & -2.4240 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0000 \\ 9.3760 & 17.1690 & 0.1340 & -49.2050 & -0.0560 & -19.8440 & 0.9250 \end{bmatrix} X \quad (3.4)$$

$$+\begin{bmatrix} 0 \\ 0 \\ -0.2570 \\ 0 \\ 0 \\ 0 \\ -0.4560 \end{bmatrix} u + \begin{bmatrix} 0.147s^2 + 1.154d^2 - 31.430c^2 + 1.582dc^2 - 2.634a^2 \\ 0 \\ -2.301s^2 - 10.863d^2 + 191.650c^2 + 0.356dc^2 + 7.019a^2 \\ 0 \\ -0.224s^2 - 1.311d^2 + 24.932c^2 + 0.186dc^2 + 0.795a^2 \\ 0 \\ -2.302s^2 - 9.271d^2 + 202.462c^2 - 0.383dc^2 \end{bmatrix}$$

$$\dot{X} = \mathbf{A}X + \mathbf{B}u + g(\text{non} - \text{linear terms})$$

3.3 Feedback Control Approach

The model is linearized about the origin to get the following system dynamics,

$$\begin{aligned} \dot{X} &= \mathbf{A}X + \mathbf{B}u \\ y = s &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} X = \mathbf{C}X \end{aligned} \tag{3.5}$$

This system is unstable as the poles are $-1.3864 \pm 2.5428i$, 0.73078, $1.4533 \pm 1.5853i$, and $0.44521 \pm 2.3142i$. The optimal gain matrix is determined with the help of the `lqr` function in MATLAB. With $\mathbf{Q} = I_{7\times7}$ and $\mathbf{R} = 1$, the following gain matrix is received,

$$K = \begin{bmatrix} 901.33 & 506.56 & 344.62 & -1059.2 & -189.45 & -828.16 & -224.57 \end{bmatrix}$$

Now, the new system dynamics equations look like,

$$\begin{aligned} \dot{X} &= (\mathbf{A} - \mathbf{BK})X + \mathbf{B}u \\ y &= \mathbf{C}X \end{aligned} \tag{3.6}$$

The poles of this system lie at $-3.699 \pm 2.3894i$, $-1.6276 \pm 0.3548i$, $-0.74503 + 0i$, and $-0.34157 \pm 2.6507i$, making the system stable. The zeros lie at $-5.6425 + 0i$, $0.47693 \pm 2.8228i$, $0.47693 - 2.8228i$, $1.2191 + 0i$, and $-0.49936 + 0i$. Figure 3.2 represents the bode plot for the closed-loop system.

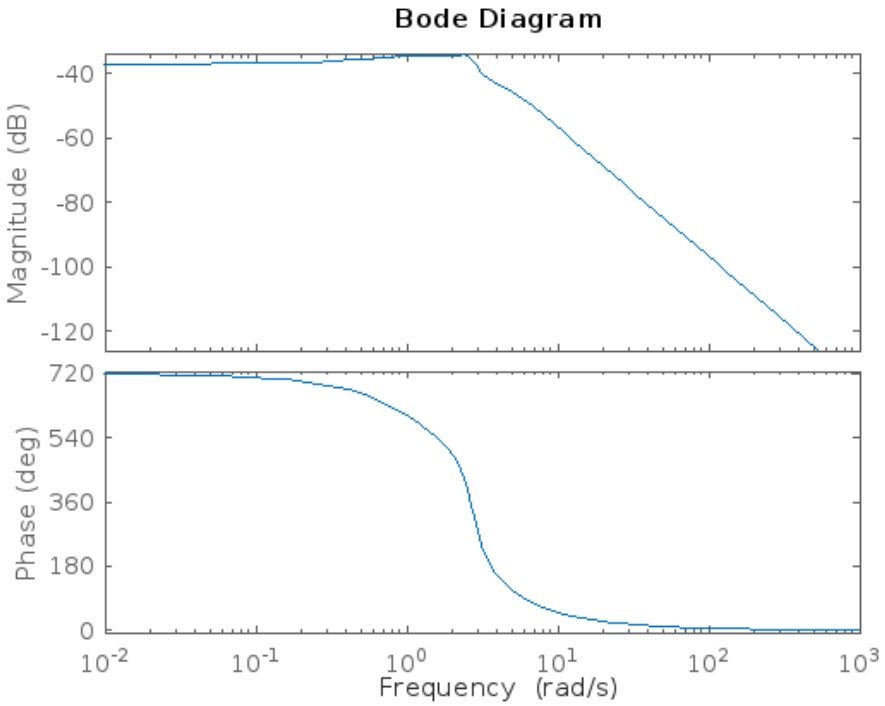


Figure 3.2: Bode Plot of closed-loop system

The goal is to track a required trajectory of slip estimation values; that is, when the object slips across the gel surface, it has slip estimation values that match that trajectory. It is evident from Figure 3.3 that tracking fails in the case of using feedback control only. The black dashed line represents the desired trajectory, and the red line indicates the predicted y values when feedback control is applied.

3.4 Feedforward Control Approach

A feedforward approach can be applied in cases where the desired trajectory is provided, along with the system dynamics. The first step is to determine the internal dynamics of the system, followed by their solution, which results in the calculation of the desired inverse input [9].

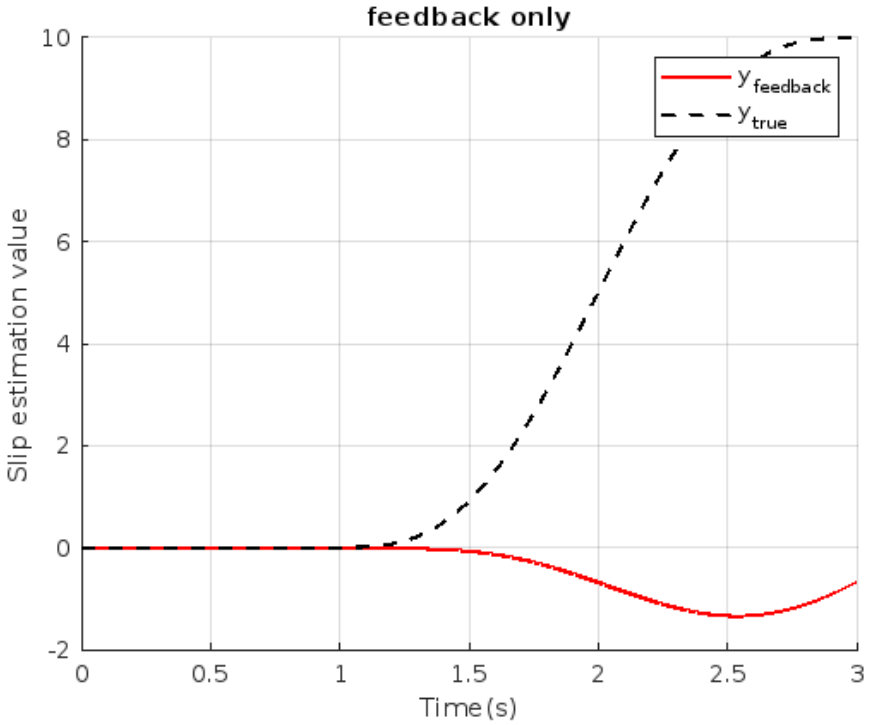


Figure 3.3: Results of tracking problem with feedback only

3.4.1 Internal Dynamics

The internal states of the system can be written as,

$$z = \mathbf{T} X_{inv} \tag{3.7}$$

This can be divided into two parts as,

$$\begin{bmatrix} \zeta \\ \hline \eta \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \hline \mathbf{T_b} \end{bmatrix} X_{inv} \tag{3.8}$$

where ζ are the known states and η are the unknown states. In order to find the unknown states, it is important to have the $\mathbf{T}$ matrix invertible. Hence, $\mathbf{T}_b$ is chosen to ensure that

the $\mathbf{T}$ matrix is full rank. This leads to the following,

$$\begin{bmatrix} \zeta \\ \text{—} \\ \eta \end{bmatrix} = \begin{bmatrix} 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.4540 & -1.8970 & -0.2360 & 7.4770 & 0.8250 & 2.7930 & 0.4520 \\ 0 & 1.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix} X_{inv}$$
$$X_{inv} = \mathbf{T}^{-1} \begin{bmatrix} \zeta \\ \text{—} \\ \eta \end{bmatrix} = \begin{bmatrix} \mathbf{T}_L^{-1} & | & \mathbf{T}_R^{-1} \end{bmatrix} \begin{bmatrix} \zeta \\ \text{—} \\ \eta \end{bmatrix} \tag{3.9}$$

Since the desired y values, referred to as y_d, are known, the desired ζ values, referred to as ζ_d, are also known. The internal dynamics can then be written as,

$$\begin{aligned} \dot{\eta}_{inv} &= \mathbf{A}_{inv}\eta_{inv} + \mathbf{B}_{inv}\mathbb{Y}_d \\ u_{inv} &= \mathbf{C}_{inv}\eta_{inv} + \mathbf{D}_{inv}\mathbb{Y}_d \end{aligned} \tag{3.10}$$

where,

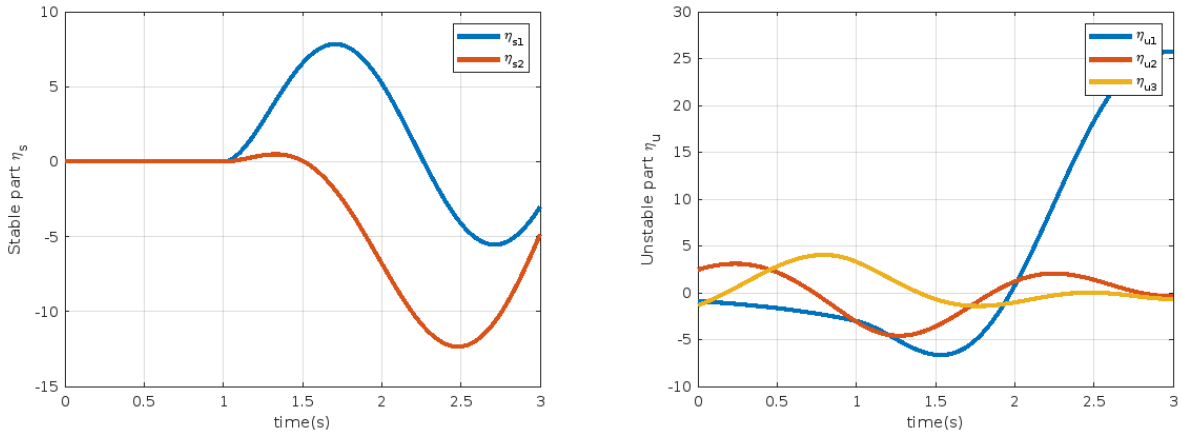
$$\begin{aligned} \mathbf{A}_{inv} &= \mathbf{T}_b \left[\mathbf{A} - \left(\mathbf{B}\mathbf{B}_y^{-1}\mathbf{A}_y\right)\right] \mathbf{T}_R^{-1} \\ \mathbf{B}_{inv} &= \begin{bmatrix} \mathbf{T}_b \left[\mathbf{A} - \left(\mathbf{B}\mathbf{B}_y^{-1}\mathbf{A}_y\right)\right] \mathbf{T}_L^{-1} & | & \mathbf{T}_B\mathbf{B}\mathbf{B}_y^{-1} \end{bmatrix} \\ \mathbf{C}_{inv} &= -\mathbf{B}_y^{-1}\mathbf{A}_y\mathbf{T}_R^{-1} \\ \mathbf{D}_{inv} &= \begin{bmatrix} -\mathbf{B}_y^{-1}\mathbf{A}_y\mathbf{T}_L^{-1} & | & \mathbf{B}_y^{-1} \end{bmatrix} \\ \mathbf{A}_y &= \mathbf{C}\mathbf{A}^2 \\ \mathbf{B}_y &= \mathbf{C}\mathbf{A}\mathbf{B} \end{aligned}$$

3.4.2 Split Dynamics

The internal states need to be decoupled into stable (η_s) and unstable (η_u) parts and then solved individually.

$$\eta_{inv} = \mathbf{T}_{split} \begin{bmatrix} \eta_s \\ \eta_u \end{bmatrix}$$
$$\begin{bmatrix} \dot{\eta}_s \\ \dot{\eta}_u \end{bmatrix} = \begin{bmatrix} \mathbf{A}_s & 0 \\ 0 & \mathbf{A}_u \end{bmatrix} \begin{bmatrix} \eta_s \\ \eta_u \end{bmatrix} + \begin{bmatrix} \mathbf{B}_s \\ \mathbf{B}_u \end{bmatrix} \mathbb{Y}_d \tag{3.11}$$

It is important to note here that the unstable internal dynamics are backward stable. The stable-unstable split dynamics are solved to determine the desired η values over the trajectory. Figure 3.4 shows the calculated values of the desired internal states.



(a) Stable internal states solution

(b) Unstable internal states solution

Figure 3.4: Internal States as a function of time

3.4.3 Inverse Input Calculation

From Equations 3.9 and 3.3, the required inverse input is calculated as,

$$u_{inv} = \frac{\begin{array}{c} -0.454\dot{y_d} - 1.897dd - 0.236\dot{dd} + 7.47dc + 0.825\dot{dc} + 2.79da + 0.45\dot{da} \\ +(2yd \times 0.147\dot{y_d}) + (1.154dd \times 2d) - (31.43dc \times 2c) + (1.582\dot{dc} \times 2dc) \\ -(2.634da \times 2a) \end{array}}{-0.236 \times 0.257 + 0.45 \times 0.456} \tag{3.12}$$

Figure 3.5 shows the calculated inverse input as a function of time.

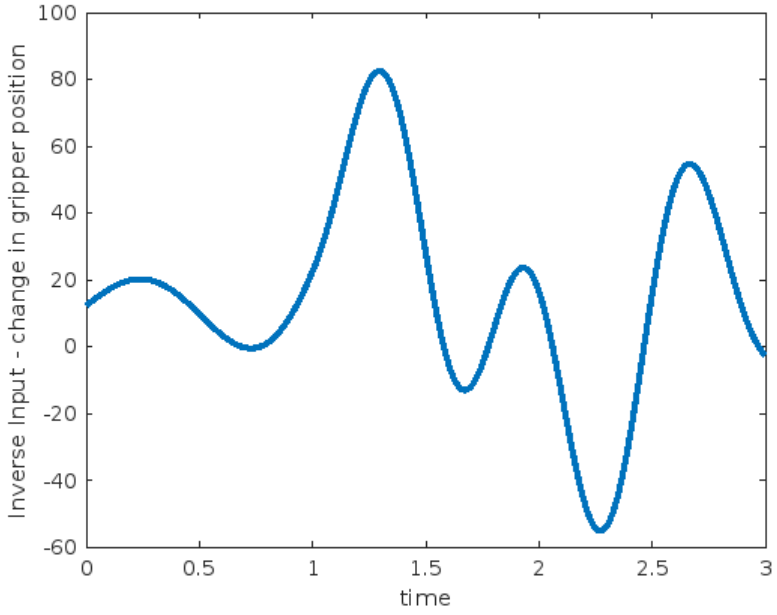


Figure 3.5: Inverse Input

3.4.4 Results

The results of using the feedforward with feedback approach can be seen in Figure 3.6. Decent tracking was achieved with the help of feedforward control.

Although the estimated values from the system overshoot the required values, toward the end, the control scheme can track the majority of the trajectory. This could be improved by

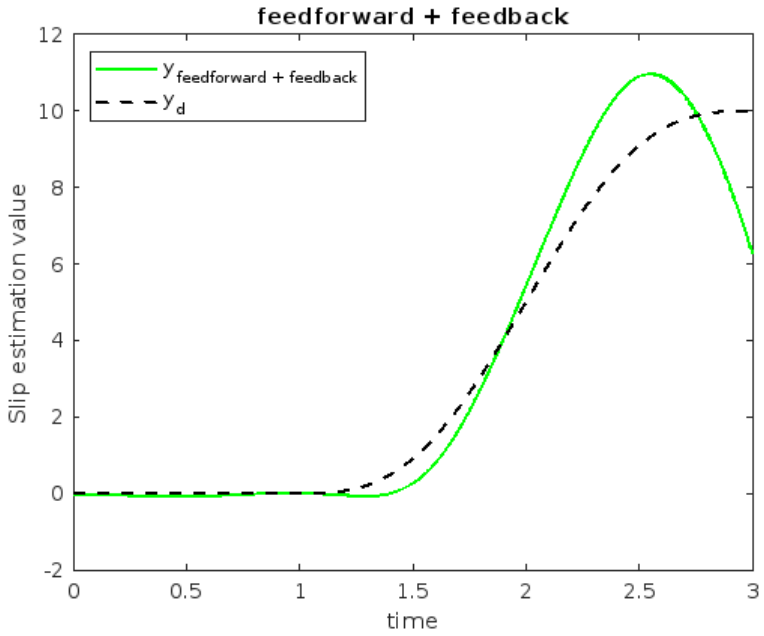


Figure 3.6: Tracking results of feedforward with feedback

using more non-linear functions to determine the system dynamics with the help of PySINDy. For this project's scope, the relations were restricted to being quadratic in nature.

Code for feedforward and feedback control and training the PySINDy model can be found at https://github.com/macs-lab/ivslip-gripper-control.

Chapter 4

OBJECT CHARACTERIZATION

4.1 Motivation

Robotic manipulation in practical applications often requires handling objects with diverse material properties, such as packaging in logistics, food items in automated kitchens, or delicate components in manufacturing [27]. In these scenarios, the ability to adaptively control grip force based on the object's characteristics is critical to ensuring both the safety and efficiency of operations.

In this setting, one of the significant challenges is not letting these objects slip while making sure the objects aren't harmed. Current methods in the industry often just rely on fixed control parameters and simple control algorithms, which can lead to model errors, such as applying too much force to delicate items or not enough to securely grip hard objects [24]. These fallacies can result in damaged goods or even mechanical failure that can increase operational costs.

To address these issues, nuanced slip control with object characterization is necessary. By characterizing objects as either deformable or hard, a robot can adjust its grip force in real-time, ensuring secure handling without causing damage. The use of advanced machine learning techniques, such as transformers paired with tactile sensor images, provides a powerful solution to this problem [21]. Transformers excel at processing complex, multi-dimensional data, allowing for accurate and dynamic classification of objects based on their tactile feedback.

Vision Transformers (ViTs) have emerged as a powerful alternative to traditional Convolutional Neural Networks (CNNs) in the field of Computer Vision and Robotics. Unlike CNNs, which rely on localized convolutional filters to capture spatial hierarchies, ViTs treat

an image as a sequence of patches and process it through a transformer model originally designed for natural language processing [26]. Each patch is embedded into a vector, and the transformer uses self-attention mechanisms to capture both local and global dependencies across the entire image. This approach allows ViTs to model long-range relationships more effectively, making them particularly suited for tasks requiring a holistic understanding of the image. Although initially designed for large datasets, recent advancements have made ViTs adaptable for smaller datasets, which is helpful because it is difficult to collect large datasets in real-world applications.

Conventional Vision Transformers (ViTs) typically require a large dataset to train the transformer's weights effectively. However, by using the method proposed by Lee et al. [14], the small tactile image dataset collected in this project can be utilized to train the object characterization pipeline. Through the application of Shifted Patch Tokenization (SPT) and Locality Self-Attention (LSA), sufficient inductive bias was introduced, enabling the transformer to learn effectively from a smaller dataset.

Moreover, I hope that this method can generalize across varieties of tactile sensors and grippers, not limiting its application to specific hardware.

4.2 Methodology

4.2.1 Data Collection and Annotation Process

Twenty objects were selected for collecting the dataset needed for the ViT, with an equal number of hard and deformable objects as shown in Figure 4.1. Each object was secured between the gripper fingers to ensure it remained stationary throughout the data collection process. Starting from the position where the tactile sensor was just in contact with the object, the gripper position was adjusted incrementally ten times, with a delay of 12 seconds between each adjustment. Tactile image frames were recorded and saved from the beginning of the experiment until the gripper reached its final position. These images were then manually annotated based on whether the object was categorized as hard or deformable. Some

examples of tactile images can be seen in Figure 4.2.

Figure 4.1: Training objects: ten hard and ten deformable objects. The deformable objects had varied levels of compressibility and elasticity.

4.2.2 Training details

Each image is then resized to be 224 x 224 so that it can be divided into 7sevenpatches. These images serve as input for the ViT model. Parameters such as image size, patch size, number of classes, dimension of the last output, depth (number of transformer blocks in the model), and number of heads in the multi-head attention layer were set to 224, 32, 3, 128, 6, and 16, respectively. The output of the model is then compared with the original label of the image, and a cross entropy loss is recorded for each epoch. The validation accuracy on train objects for ten epochs was 92.41%.

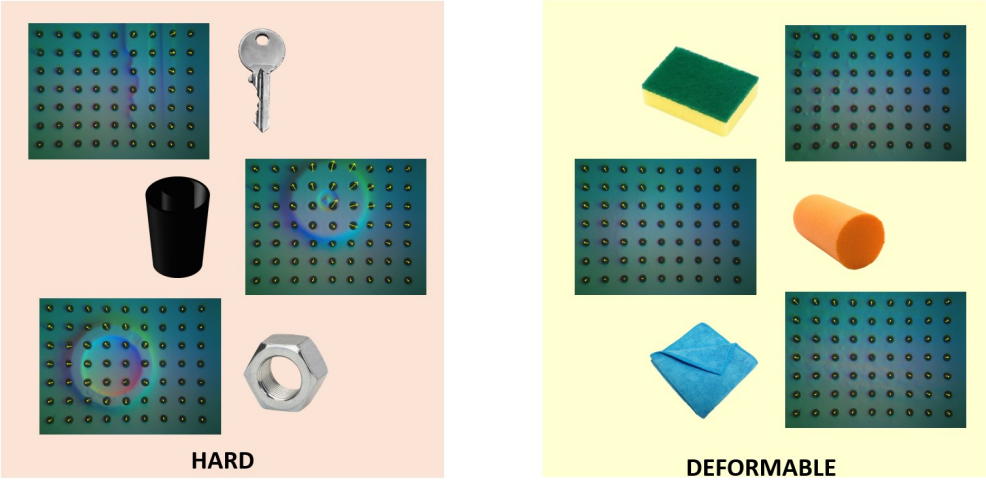


Figure 4.2: Tactile frame image examples and their labels

4.3 Results

The trained ViT model was then tested on ten unseen objects, as shown in Figure 4.3. The confusion matrix for these tests is shown in Figure 4.4.

Figure 4.3: Test objects for ViT: five hard and five deformable objects

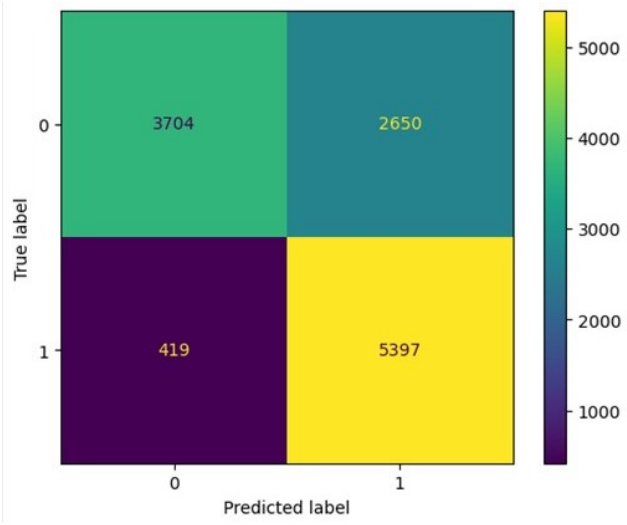


Figure 4.4: Confusion matrix analysis

It is important to note that the accuracy of this model on unseen data was 77.4%, and the failure cases are evident in the confusion matrix. This can be attributed to the fact that cylindrical objects, such as steel rods and PVC pipes, impart a soft outline on the gel surface due to their curved surfaces. In contrast, objects like the thread roll and Velcro roll have sharp edges, which can cause them to be labeled as hard objects, even though they do deform under applied pressure. To improve the model's performance, the numerical data collected from the tactile features could be concatenated with the images after they have been divided into patches. This approach would ensure that important features like the divergence and curl of the deformation field are also provided to the ViT.

Code for object characterization using vision transformer can be found at https://github.com/macs-lab/slip-object-classification.

4.4 Gripper Force Control

The confidence score, denoted as C, can be used as a metric to decide the normal force the gripper fingers apply on the objects grasped. The confidence score of a neural network

quantifies its certainty in a prediction, represented as a probability between 0 and 1. It reflects how likely the network believes a given input belongs to a particular class. Two threshold values can be set for this purpose: the lower confidence score value is denoted as C_{low}, and the higher value is denoted as C_{high}. The normal force applied by the gripper is denoted as G. For hard objects, the following rule can be applied,

$$G = \begin{cases} G_{strong} & if\ C > C_{high} \\ G_{light} & if\ C_{high} > C > C_{low} \\ G_{low} & if\ C_{low} > C \end{cases} \tag{4.1}$$

For soft objects, the normal force can be determined as,

$$G = \begin{cases} G_{light} & if\ C > C_{high} \\ G_{low} & if\ C_{low} > C \end{cases} \tag{4.2}$$

This ensures that according to the ViT model's result, the force applied can be controlled to ensure that each type of object is handled within its tolerance range. This also incorporates the chance of failure by the transformer model by using the confidence score as a metric.

Chapter 5

CONCLUSION

This research has successfully built two control schemes for a two-finger gripper. The first scheme controls the change in the position of the gripper fingers to achieve the desired slip severity estimation, and the second scheme controls the normal force applied to the object based on vision-based object characterization. The key contributions of this research are:

- Improved the slip detection method based on tactile feature extraction.
- Proposed a feedforward with feedback control scheme to achieve desired slip velocity.
- Proposed a vision-based control scheme to grasp hard and deformable objects without damaging them.

BIBLIOGRAPHY

[1] Roberto Calandra, Andrew Owens, Manu Upadhyaya, Wenzhen Yuan, Justin Lin, Edward H. Adelson, and Sergey Levine. The feeling of success: Does touch sensing help predict grasp outcomes?, 2017.

[2] Wei Chen, Heba Khamis, Ingvars Birznieks, Nathan F. Lepora, and Stephen J. Redmond. Tactile sensors for friction estimation and incipient slip detection—toward dexterous robotic manipulation: A review. *IEEE Sensors Journal*, 18(22):9049–9064, 2018.

[3] Brian M. de Silva, Kathleen Champion, Markus Quade, Jean-Christophe Loiseau, J. Nathan Kutz, and Steven L. Brunton. Pysindy: A python package for the sparse identification of nonlinear dynamics from data, 2020.

[4] Siyuan Dong, Daolin Ma, Elliott Donlon, and Alberto Rodriguez. Maintaining grasps within slipping bound by monitoring incipient slip, 2018.

[5] GelSight. Gelsight mini tactile sensor. Accessed: 2024-08-10.

[6] Abhinav Gupta and Larry S. Davis. Objects in action: An approach for combining action understanding and object perception. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[7] 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.

[8] R.D. Howe, N. Popp, P. Akella, I. Kao, and M.R. Cutkosky. Grasping, manipulation, and control with tactile sensing. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 1258–1263 vol.2, 1990.

[9] D. Iamratanakul, H. Perez, and S. Devasia. Feedforward trajectory design for output transitions in discrete-time systems: disk-drive example. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 4, pages 3142–3147 vol.4, 2003.

[10] Jasper Wollaston James, Nicholas Pestell, and Nathan F. Lepora. Slip detection with a biomimetic tactile sensor. *IEEE Robotics and Automation Letters*, 3(4):3340–3346, 2018.

[11] Roland S. Johansson. *Sensory Input and Control of Grip*, pages 45–63. John Wiley Sons, Ltd, 2007.

[12] Mohsen Kaboli, Kunpeng Yao, and Gordon Cheng. Tactile based manipulation of deformable objects with dynamic center of mass. 10 2016.

[13] Raj Kolamuri, Zilin Si, Yufan Zhang, Arpit Agarwal, and Wenzhen Yuan. Improving grasp stability with rotation measurement from tactile sensing. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 6809–6816. IEEE Press, 2021.

[14] Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. Vision transformer for small-size datasets, 2021.

[15] Nathan F Lepora, Uriel Martinez-Hernandez, and Tony J Prescott. Active bayesian perception for simultaneous object localization and identification. In *Robotics: Science and Systems*, pages 1–8, 2013.

[16] Mingxuan Li, Yen Hang Zhou, Tiemin Li, and Yao Jiang. Incipient slip-based rotation measurement via visuotactile sensing during in-hand object pivoting, 2024.

[17] Yi Li, Muru Zhang, Markus Grotz, Kaichun Mo, and Dieter Fox. STOW: Discrete-frame segmentation and tracking of unseen objects for warehouse picking robots. In *7th Annual Conference on Robot Learning*, 2023.

[18] C. Melchiorri. Slip detection and control using tactile and force sensors. *IEEE/ASME Transactions on Mechatronics*, 5(3):235–243, 2000.

[19] Andres Ospina, Saifeddine Aloui, Christelle Godin, Mathieu Grossard, and Alain Micaelli. Tactile sensing system for robotics dexterous manipulation based on a matrix of 3-axes force sensors. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 2105–2110. IEEE Press, 2015.

[20] Maciej Pajak and Antje Nuthmann. Object-based saccadic selection during scene perception: Evidence from viewing position effects. *Journal of Vision*, 13(5):2–2, 04 2013.

[21] En Yen Puang, Zechen Li, Chee Meng Chew, Shan Luo, and Yan Wu. Learning stable robot grasping with transformer-based tactile control policies, 2024.

[22] Robotiq. Hand-e adaptive robot gripper, n.d. Accessed: 2024-08-15.

[23] Universal Robots. Ur5 robot, n.d. Accessed: 2024-08-15.

[24] G. P. Slota, M. L. Latash, and V. M. Zatsiorsky. Grip forces during object manipulation: experiment, mathematical model, and validation. *Experimental Brain Research*, 213(1):125–139, 2011.

[25] Marc R. Tremblay and Mark R. Cutkosky. Estimating friction using incipient slip sensing during a manipulation task. *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 429–434 vol.1, 1993.

[26] Cong Wang, Hongmin Xu, Xiong Zhang, Li Wang, Zhitong Zheng, and Haifeng Liu. Convolutional embedding makes hierarchical vision transformer stronger. *ArXiv*, abs/2207.13317, 2022.

[27] Z. Wang, S. Hirai, and S. Kawamura. Challenges and opportunities in robotic food handling: A review. *Frontiers in Robotics and AI*, 8:789107, 2022.