A Novel Three-Finger Dexterous Hand with Visual-Based Grasp Planning and Tactile-Based Stable Grasping

von Tao Wang

Dissertation

zur Erlangung des Grades eines Doktors der Ingenieurwissenschaften
- Dr.-Ing. -

Vorgelegt im Fachbereich 3 (Mathematik & Informatik)
der Universität Bremen
im August 2023

Datum des Promotionskolloquiums: 19. Dezember 2023

Gutachter:	Prof. Dr. Frank Kirchner	(Universität Bremen)
	Prof. Dr. Udo Frese	(Universität Bremen)
	Prof. Dr. Joachim Hertzberg	(Universität Osnabrück)

Abstract

Robotic manipulation is a complex field that still faces numerous challenges. With the development of collaborative robots, the end-effector area is just beginning to gain traction, and electrically driven end-effectors are just starting to be widely used in the industry. One of the fundamental requirements for a robot to achieve practical applications is stable grasping. However, in reality, most stable grasping relies on specific static conditions and human experience. As the robotic industry continues to develop more complex and diverse applications, the need for stable grasping to support these high-level applications increases. To achieve more complex and stable grasping functions, a complex end-effector, such as a dexterous hand, becomes essential hardware.

This thesis focuses on building a novel practical dexterous hand that can be used in robotic manipulation research. To make better use of the dexterous hand, visual-based grasp planning, and tactile-based stable grasping are necessary to form a stable grasping system. In this thesis, three crucial topics were selected and divided into three parts of the work: dexterous hand, grasp planning, and stable grasping.

The dexterous hand part includes the main design work of a modular three-finger dexterous hand called DoraHand and partial work on another one called Eagle Shoal. The performance of the DoraHand and the tactile sensor module is showcased through experiments. Two-finger and five-finger versions of the DoraHand have also been developed and tested in real applications, providing a reliable hardware foundation for further research.

The grasp planning part focuses on providing a grasp planning solution for the dexterous hand. As an essential function of using an end-effector, this part starts with an analytic solution that considers the limitations of the dexterous hand mechanism and grasp quality evaluation. A grasp planning network has been developed using both analytic and data-driven approaches. The network features a multi-finger grasp plan representation method and has been successfully verified.

The stable grasping part is the final application of this thesis, where the hardware provides the foundation and the stable grasping algorithm utilizes the tactile sensor. An open-source visual-tactile dataset has been developed using the Eagle Shoal dexterous hand. The stable grasping algorithm, built based on this dataset, has been successfully verified with different types of end-effectors, including DoraHand and suction cup gripper.

Overall, these three parts of work constitute the critical components of a stable grasping system using a dexterous hand. This system and related dataset enable further research in stable grasping and robotic manipulation.

The primary objective of this thesis has been successfully achieved with the development of DoraHand, which has been used by over twenty research institutes and companies. The algorithms developed for grasp planning and stable grasping serve as a foundation for future research in this field, while the dataset can be used as a benchmark for comprehensive robotic research. Further development is needed to explore the potential applications of the dexterous hand in robotic manipulation.

Zusammenfassung

Robotermanipulation ist ein komplexes Gebiet, das noch vor vielen Herausforderungen steht. Mit der Entwicklung kollaborativer Roboter gewinnt der Bereich der Endeffektoren gerade erst an Dynamik, und elektrisch angetriebene Endeffektoren werden in der Industrie gerade erst in großem Umfang eingesetzt. Eine der grundlegenden Anforderungen an einen Roboter für praktische Anwendungen ist das stabile Greifen. In der Realität hängt das stabile Greifen jedoch meist von spezifischen statischen Bedingungen und menschlicher Erfahrung ab. Da die Roboterindustrie immer komplexere und vielfältigere Anwendungen entwickelt, steigt der Bedarf an stabilem Greifen zur Unterstützung dieser anspruchsvollen Anwendungen. Um komplexere und stabilere Greiffunktionen zu erreichen, ist ein komplexer Endeffektor, wie zum Beispiel eine geschickte Hand, eine wesentliche Hardwarekomponente.

In dieser Arbeit geht es um den Bau einer neuartigen, praktischen, geschickten Hand, die in der Forschung zur Robotermanipulation eingesetzt werden kann. Um die geschickte Hand besser nutzen zu können, sind eine visuell basierte Greifplanung und ein taktil basiertes, stabiles Greifsystem unerlässlich, um ein stabiles Greifsystem zu bilden. In dieser Arbeit wurden drei entscheidende Themen ausgewählt und in drei Teile unterteilt: geschickte Hand, Greifplanung und stabiles Greifen.

Der Teil geschickte Hand umfasst den Hauptentwurf einer modularen geschickte Hand mit drei Finger namens DoraHand und Teilarbeiten an einer anderen Hand namens Eagle Shoal. Die Leistungsfähigkeit der DoraHand und des taktilen Sensormoduls wird durch Experimente demonstriert. Versionen der DoraHand mit zwei- und fünf Finger wurden ebenfalls entwickelt und in realen Anwendungen getestet und bieten eine zuverlässige Hardware-Basis für weitere Forschungen.

Der Teil zur Greifplanung konzentriert sich auf die Bereitstellung einer Lösung zur Planung des Greifens mit der geschickten Hand. Da die Verwendung eines Endeffektors eine wesentliche Funktion darstellt, beginnt dieser Teil mit einer analytischen Lösung, die die Einschränkungen des Mechanismus der geschickten Hand und die Bewertung der Qualität des Greifsystems berücksichtigt. Mit Hilfe von analytischen und datengesteuerten Ansätzen wurde ein Griffplanungsnetz entwickelt. Das Netzwerk verfügt über eine Methode zur Darstellung einer Planung des Greifens mit mehreren Fingern und wurde erfolgreich überprüft.

Der Teil zum stabilen Greifen ist die letzte Anwendung dieser Arbeit, bei der die Hardware die Grundlage bildet und der stabile Algorithmus des Greifsystems den taktilen Sensor nutzt. Es wurde ein visueller taktiler Open-Source-Datensatz entwickelt, der die geschickte Hand von Eagle Shoal verwendet. Der stabile Algorithmus des Greifsystems, der auf diesem Datensatz basiert, wurde erfolgreich mit verschiedenen Arten von Endeffektoren überprüft, einschließlich der DoraHand und Saugnapf-Greifer.

Insgesamt bilden diese drei Einzelteile der Arbeit die entscheidenden Komponenten eines stabilen Greifsystems mit einer geschickten Hand. Dieses System und der zugehörige Datensatz ermöglichen weitere Forschungen zum stabilen Greifen und zur Robotermanipulation.

Das Hauptziel dieser Arbeit wurde mit der Entwicklung von DoraHand erfolgreich erreicht, das bereits von über zwanzig Forschungsinstituten und Unternehmen verwendet wird. Die entwickelten Algorithmen zur Greifplanung und zum stabilen Greifen dienen als Basis für zukünftige Forschungen auf diesem Gebiet, während der Datensatz als Benchmark für umfassende Roboterforschung verwendet werden kann. Weitere Entwicklungen sind notwendig, um die möglichen Anwendungen der geschickten Hand bei der Robotermanipulation zu erforschen.

Danksagung

The work presented in this thesis has been a long-term endeavor spanning over the past five years, during which time I changed jobs from Intel Labs China to Dorabot. I am fortunate that my supervisor continued to support me and the work content continued in both of these companies. Despite experiencing three years of special circumstances, with the support of various people, all the work related to this thesis progressed smoothly. I would like to take this opportunity to express my gratitude and appreciation to those who have helped me throughout this journey.

First and foremost, I would like to thank my first supervisor Prof. Frank Kirchner. Prof. Frank accepted my application in Jan 2018 and has always encouraged and supported my direction over the past years. Even when my work was blocked in 2021, he continued to support and provide valuable advice, enabling me to complete this Ph.D.. I am grateful for his unwavering support and guidance throughout the thesis.

I would like to express my gratitude to my second supervisor Prof. Udo Frese at the University of Bremen, for accepting the role and providing helpful suggestions during my Ph.D. thesis. And I would like to thank Prof. Joachim Hertzberg, who is the third reviewer of my Ph.D. thesis and give me valuable feedback. I would also like to thank the committee professor, Prof. Rolf Drechsler, who provided support with the colloquium.

A special thanks to colleagues at DFKI RIC, especially Bingbin Yu, who supported me in Bremen and helped me complete some material work, and Petra who provided invaluable assistance throughout the procedure and material preparation. I would also like to thank Shivesh Kumar and Rohit Menon, who provided support and encouragement during my stay in Bremen and helped me with my paper. Thanks to Manuel Meder, who provided support with the colloquium.

I would like to express my gratitude to Intel Labs China for giving me the opportunity to start my research in the robotic manipulation area. I am thankful to the lab's director, Jiqiang Song, for accepting me and supporting my Ph.D. journey. I would also like to thank my team leader, Hong Zhang, for encouraging me to pursue my Ph.D. and helping me connect with DFKI. I would like to thank my friend and colleague Bo Kang for encouraging me on the application and supporting the work of Eagle Shoal. Additionally, I would like to thank my colleagues Zhanxiao Geng, Xiaochuan Luo, and Yongfa Zhou for their support in the work of Eagle Shoal.

I would like to thank Dorabot for supporting me in continuing my work on the dexterous hand and delivering the DoraHand product. I am grateful to CEO Yaohuan Deng for supporting me in building a team in the robotic manipulation direction, and to CTO Hao Zhang for supporting and encouraging me to continue the direction and the product. I would also like to thank team members Sven Chen, Rex Xie, Yuan Li, Yan Zhang, and Yulun, Xie for their direct support in product iteration and project management.

I am grateful to my postgraduate supervisor, Prof. Jihong Yan, for guiding me to begin my research work and supporting my Ph.D. application. I would also like to thank Prof. Bing Fang and Prof. Huaping Liu for providing opportunities for me to

have better communication with the research area. Finally, I would like to express my appreciation to Prof. Guanjun Bao for his support in the research area.

Lastly, I would like to express my gratitude to my wife, Zhiying Fang, for supporting me throughout my life and encouraging me to overcome all the challenges along the way. I am also thankful for the support of my daughter Xi Wang, who motivates me to move forward better. Finally, I would like to thank all my family members and friends for their unwavering support.

Shenzhen, im Dezember 2023 Tao Wang

Contents

Part I

Introduction and State of the Art

Chapter 1

Introduction

This chapter serves as the entry point of this thesis and is divided into five primary sections. Section 1.1 outlines the motivation behind the thesis and illustrates the relevance of the main topics addressed. Section 1.2 establishes the primary objectives of this thesis. Section 1.3 highlights the significant contributions that this project makes to the field. Section 1.4 introduces the primary structure of this thesis. Section 1.5 summarizes the outcomes and results of this thesis.

1.1 Motivation

The manipulation function for a universal robot is crucial. However, achieving this remains a challenge due to several factors, such as complex hardware, sensing capabilities, dynamic control, multi-level planning, unstructured environments, and others. One fundamental requirement within this domain is achieving stable grasping, which remains a significant challenge in the field of robotics.

Figure 1.1 presents a diagram of a robotic manipulation system that integrates essential hardware and software components. The hardware system should include robot base, robotic arm, end-effector, controller, vision sensor, and other sensors. Meanwhile, the software system should encompass motion control, arm planner, grasp planner, task planner, visual processor, and other sensor processors.

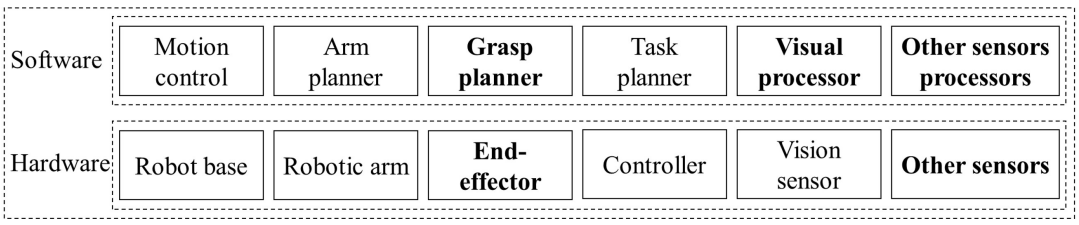


Figure 1.1: Structure of a stable grasping system

The main motivation behind this thesis is to enhance robotic manipulation through the development of dexterous hand hardware and related software. To achieve this goal, several critical components have been identified and labeled in Figure 1.1. These components include the end-effector, other sensors, grasp planner, visual sensor processor, and other sensor processors. These components can be grouped into three primary areas of focus: 1.1.1 dexterous hand, 1.1.2 grasp planning, and 1.1.3 stable grasping. By addressing these key areas, this thesis introduces a system with a novel dexterous hand hardware, a visual-based grasp planning component, and a tactile-based stable grasping component. This thesis aims to make contributions to the field of robotic manipulation with such a system.

1.1.1 Dexterous Hand

An end-effector is a critical component for accomplishing dexterous tasks, and a dexterous hand end-effector can provide even greater capabilities. Drawing inspiration from the human hand, which plays a vital role in human bodies, serves as a fundamental direction for designing a dexterous hand.

The structure and Degrees of Freedom (DOF) of the human hand are illustrated in Figure 1.2 [ElKoura and Singh, 2003]. The hand has a total of 27 DOF: 5 DOF in the thumb, 2 for extension and flexion, and 3 for the thumb base joint; 4 DOF in each finger, 3 for extension and flexion, and 1 for abduction and adduction; and 6 DOF in the wrist, including rotation and translation [Agur and Dalley, 2009]. Designing a dexterous hand with the same number of DOF and structure can help achieve similar dexterity for the robot, but it is not an easy task due to limitations such as actuator, material, lifetime, maintenance, etc.

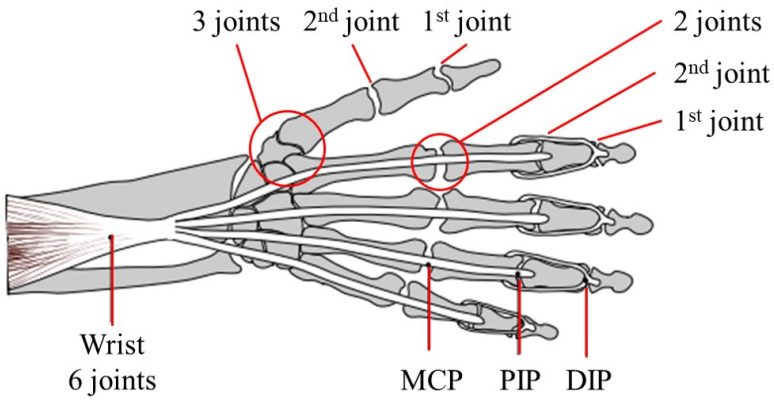


Figure 1.2: Bones and joints of the hand

Designing a dexterous hand for stable grasping does not necessarily require as many DOF as a human hand. To achieve stable grasping and perform certain manip-

ulation tasks, a dexterous hand that is both user-friendly and sufficiently complex is necessary. In addition to motion capability, sensing capability is also important and can enhance the dexterity of the hand. Referring to the human hand, tactile and joint sensing become essential components, augmenting the hand's overall capabilities. Furthermore, maintenance and cost should be limited to enable widespread use in research areas. Thus, one of the motivations of this work is to help to lower hardware barriers, allowing researchers to focus on the other areas that have not been well studied. Such work needs to find a balance between the requirements of DOF, sensing capability, maintenance, cost, and others.

1.1.2 Grasp Planning

Grasping in daily life is diverse, and while human grasp strategies may seem natural, they vary depending on the object being grasped. To understand the detailed process of grasping an object to complete a transport or manipulation task, it can be divided into three stages as shown in Figure 1.3 [Chang et al., 2010]. The pre-grasp manipulation stage involves target sensing and grasp planning based on input data before grasping. The grasp acquisition stage involves interacting with the object and surroundings to achieve stable grasping. The post-grasp manipulation stage involves maintaining stable grasping and completing the manipulation task, often requiring dynamic stability during the process. The focus of grasp planning is on the pre-grasp manipulation stage.

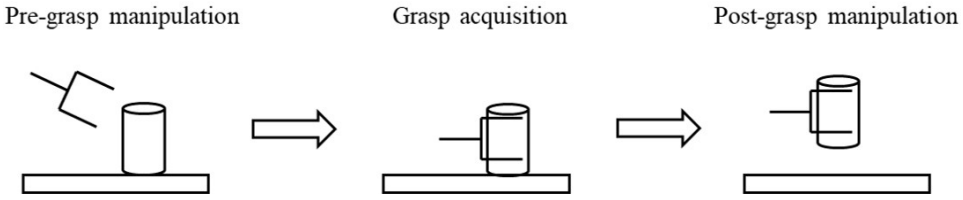


Figure 1.3: Three stages of grasp

Current work on grasp planning includes two primary approaches: analytic and data-driven [Bohg et al., 2013]. Analytic approaches have been extensively studied for various grasping cases in specifically defined environments, but are not widely used in complex scenes. Data-driven approaches are increasingly popular due to advancements in deep learning, with most works focusing on simple end-effectors such as parallel grippers and suction cup grippers.

Considering the complex control of the dexterous hand, more parameters need to be trained in a model, making it difficult to achieve good results. An analytic approach may help improve training efficiency. Combining analytic and data-driven solutions may leverage the advantages of both methods and improve the overall per-

formance. Considering the characteristics of the dexterous hand, more works needs to be explored compared to solutions for simple end-effectors.

1.1.3 Stable Grasping

In a grasping task as mentioned in Figure 1.3, the second and third stages heavily rely on sensing capabilities. For humans, ensuring stable grasping, lifting the target, and adjusting force and pose according to weight or external forces are natural processes that involve many steps, especially in sensor data processing.

Human possess several types of sensing capabilities, including visual sensing, which provides the most critical human sensing data; audio sensing, which plays a secondary role; and tactile sensing, which is crucial among other types of sensing [Boff and Lincoln, 1988]. By leveraging these sensing capabilities, humans can perform grasp-related tasks with high quality. In robotics, visual processing is the most popular area for providing target information, guiding the robot, and aiding manipulation; audio processing is a more mature area, and the key issue is the Natural Language Processing (NLP) part, and most robots are controlled by the programs without audio; tactile processing is not widely used currently but can provide more information for manipulation.

In a stable grasping task, the visual data is often occluded during grasping, the tactile data with more sensitive features play a more important role. Existing work related to tactile-based stable grasping relies on specific hardware. However, due to the universality of grasping stability, utilizing learning methods and sufficient data may provide solutions that can be easily applied to different hardware.

1.2 Objectives

The main objective of this thesis is to develop a user-friendly and practical dexterous hand, related grasp planning, and stable grasping algorithms, and contribute to the field of stable grasping in robotic manipulation. Based on the motivation discussed earlier, three specific objectives have been identified for this work.

1. **A modular dexterous hand.** Develop a user-friendly and practical modular dexterous hand equipped with sufficient DOF and tactile sensing capability for robotic manipulation research. The design should consider the layout and main components to ensure user-friendliness. The modular design and tactile sensing module are key features.

2. **Grasp planning with visual data.** Develop a grasp planning solution that uses visual data input and takes advantage of analytic and data-driven ap-

proaches to realize grasp planning for a dexterous hand. This solution utilizes more DOF of the dexterous hand.

3. **Stable grasping with tactile data.** Develop a stable grasping solution that uses tactile data input to predict grasp stability and realize stable grasping with a dexterous hand. This includes building a visual-tactile dataset and developing a stable grasping solution that utilizes this dataset.

1.3 Contributions

With the objectives mentioned in the previous section, this thesis project has been developed and has contributed to three main areas: dexterous hand design, visual-based grasp planning, and tactile-based stable grasping.

The first major contribution of this thesis is the development of a novel modular dexterous hand product. This work includes two designs of the dexterous hand: Eagle Shoal[Wang et al., 2019a] and DoraHand[Wang et al., 2022]. This thesis mainly introduced a fully-actuated modular three-finger dexterous hand, including a tactile sensor module that can help provide an affordable dexterous hand for manipulation research. Two-finger and five-finger hands were also developed using the finger module. This work has resulted in final products that are used in more than twenty universities and research institutes.

The second major contribution of this thesis is the development of a novel grasping planning method for the dexterous hand. This method uses contact points instead of a grasp box to present the grasp plan and combines analytic and data-driven approaches to generate a grasp plan tailored to the design of the dexterous hand. It has been successfully employed in real-world scenarios and verified through experimentation.

The third major contribution of this thesis is the development of a visual-tactile dataset and a grasp stability prediction method for robotic manipulation. This visual-tactile dataset included the Red, Green, and Blue (RGB) data, Depth data, tactile data, and joint data simultaneously [Wang et al., 2019b]. It has been valuable for researchers without access to hardware and can assist with simulation work. Using this dataset, stable grasping has been achieved through the development of a grasp stability evaluation method [Wang and Kirchner, 2021]. The stability prediction method employs a neural network to achieve high performance stability and can generalize to the field of time-series data.

In conclusion, the contributions of this thesis can serve as a valuable resource for researchers in the field of robotic grasping by providing comprehensive work that integrates hardware, datasets, and algorithms. Figure 1.4 summarizes the main

achievements of this work, which can facilitate future developments in robotic manipulation.

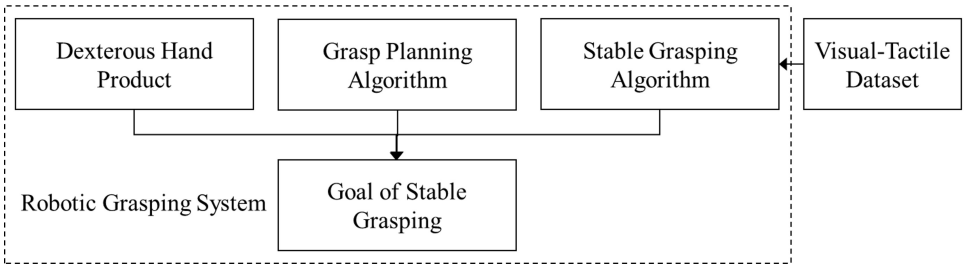


Figure 1.4: Contributions of the thesis

1.4 Structure of the Thesis

This thesis is organized into five parts: Part I Introduction and State of the Art, Part II Dexterous Hand, Part III Grasp Planning, Part IV Stable Grasping, Part V Conclusion and Outlook. Figure 1.5 shows the overall structure of this thesis.

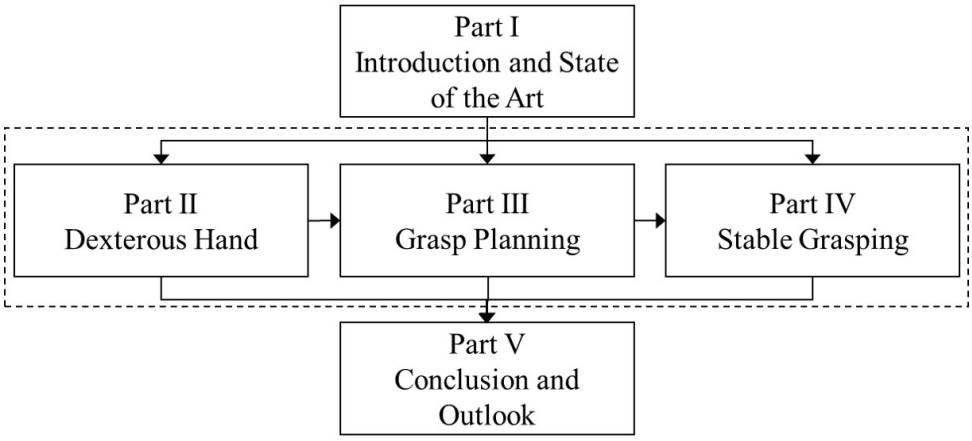


Figure 1.5: Structure of the thesis

Part I Introduction and State of the Art This part comprises two chapters that primarily focus on the introduction and state of the art. **Chapter 1 Introduction** provides the motivation, objectives, contributions, structure of the thesis, and dissemination of scientific results. The **Chapter 2 State of the Art** provides an overview of the state of the art in three different areas: dexterous hand, grasp planning, and stable grasping.

Part II Dexterous Hand This part focuses on the design and evaluation of a three-finger dexterous hand and consists of two chapters. **Chapter 3 Dexterous Hand Design** outlines the design concept, key components mechanism, and whole system design. These sections provide a comprehensive overview of the dexterous hand design process. **Chapter 4 Experiment and Performance Evaluation** covers basic performance, sensing capability, and extension applications. These results demonstrate the capability of the dexterous hand through suitable evaluation methods.

Part III Grasp Planning This part focuses on the development of grasp planning for the dexterous hand using a combination of two different approaches: analytic and data-driven. The resulting technology enables the dexterous hand to perform grasping tasks. This part comprises two chapters. **Chapter 5 Analytic Grasp Planning for Three-Finger Dexterous Hand** presents the analytic analysis and algorithm for the three-finger dexterous hand. **Chapter 6 Visual-Based Grasp Planning** explains the combined solution process of analytic and data-driven approaches and shows application results.

Part IV Stable Grasping This part focuses on stable grasping with tactile data and covers the development of a dataset and grasp stability prediction. This part comprises two chapters. **Chapter 7 Visual-Tactile Dataset** introduces details about the visual-tactile dataset. **Chapter 8 Tactile-Based Stable Grasping** explains the stable grasping algorithm based on the dataset and showcases the final results of the stable grasping tasks accomplished by the project.

Part V Conclusion and Outlook This part contains the final chapter, **Chapter 9 Conclusion and Outlook**. This chapter serves as a conclusion and outlook of the thesis by summarizing accomplished work and providing conclusions related to the hardware, dataset, and algorithms. Additionally, this section provides an outlook for the entire project.

1.5 Dissemination of Scientific Results

The scientific results presented in this thesis have been disseminated through various channels, including 1.5.1 journals, 1.5.2 conferences, 1.5.3 posters, 1.5.4 invited guest lectures, 1.5.5 awards and nominations, and 1.5.6 patents.

1.5.1 Journals

1. Wang Tao, Yang Chao, Kirchner Frank, Du Peng, Sun Fuchun, and Fang Bin (2019). Multimodal grasp data set: A novel visual–tactile data set for robotic manipulation. International Journal of Advanced Robotic Systems, 16(1), 1729881418821571.

2. Wang Tao, Xie Zheng, Li Yuan, Zhang Yan, Zhang Hao, and Kirchner Frank (2022). DoraHand: A Novel Dexterous Hand with Tactile Sensing Finger Module. Industrial Robot, 49(4), pp. 658-666.

1.5.2 Conferences

1. Wang Tao, Geng Zhanxiao, Kang Bo, and Luo Xiaochuan (May, 2019). Eagle Shoal: A new designed modular tactile sensing dexterous hand for domestic service robots. In 2019 IEEE International Conference on Robotics and Automation (ICRA), pp. 9087-9093.

2. Wang Tao, and Kirchner Frank (July, 2023). Grasp stability prediction with time series data based on STFT and LSTM. In 2023 IEEE International Conference on Advanced Robotics and Mechatronics (ICARM).

1.5.3 Posters

1. Wang Tao, Gong Shandong, Salman Yusuf, Chen Shuqu, Zhang Hao, Kirchner Frank (Nov, 2019). Design a Dexterous Hand for the Logistic Robot in Bin Picking. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Late breaking results posters.

1.5.4 Invited Guest Lectures

1. Wang Tao (8 Nov, 2019). Stable grasping in the field of logistics. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop, Manipulation through Contacts: Bridging the Gap between Research Community and Industry.

2. Wang Tao (12 May, 2021). Robotic Technologies in Logistics. Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen, China.

1.5.5 Awards and Nominations

1. 3rd place, In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2nd Robotic Grasping and Manipulation Competition, Service Robot Track, Vancouver, British Columbia, Canada.

2. Best Paper Award in Service Robotics - Finalist, In 2019 IEEE International Conference on Robotics and Automation (ICRA), Montreal, Canada.

3. DIA Silver, DoraHand, 2020, Design Intelligence Award.

4. Winner, DoraHand-F3, 2021, Red Dot Design Award.

5. Winner, DoraHand-3F, 2022, iF Design Award.

1.5.6 Patents

1. Wang Tao, Kang Bo, Robotic finger, USD829249S1, United States.

2. Wang Tao, Xie Zheng, Multi-freedom overload-proof multi-finger dexterous hand, CN114734469A, China.

3. Wang Tao, Xie Zheng, Modularized dexterous five-finger hand, CN216372242U, China.

4. Wang Tao, Chen Shuqu, Xu Yi, Dexterous hand finger and multi-finger dexterous hand, CN211362291U, China.

5. Wang Tao, Chen Shuqu, Force sensor, force sensing device, method for measuring and calculating force, and storage medium, CN110749393A, China.

6. Wang Tao, Chen Shuqu, A force sensing structure, dexterous hand finger and multi-finger dexterous hand, CN110640775A, China.

7. Wang Tao, Chen Shuqu, Force sensing structure, dexterous hand finger and multi-finger dexterous hand, CN211362292U, China.

8. Wang Tao, Chen Shuqu, Xu Yi, Dexterous hand finger and multi-finger dexterous hand, CN110640776A, China.

9. Wang Tao, Chen Shuqu, Force sensor based sliding prediction method and device, electronic equipment and storage medium, CN110793693A, China.

10. Wang Tao, Xie Zheng, Li Yuan, Chen Shuqu, Huang Zhongying, Embedded system for dexterous hand, US20230173684A1, United States.

11. Xie Zheng, Wang Tao, Finger-detachable dexterous robotic hand, WO2021223631A1, PCT.

12. Xie Zheng, Wang Tao, Dexterous hand with detachable fingers, CN212825430U, China.

13. Xie Zheng, Wang Tao, A dexterous hand with detachable fingers, US20230166410A1, United States.

14. Chen Shuqu, Wang Tao, Xie Zheng, Li Yuan, Huang Zhongying, Dexterous hand embedded system, CN212919403U, China.

Chapter 2

State of the Art

This chapter provides an overview of the state of the art in stable grasping systems and is divided into three distinct sections. Section 2.1 provides a survey of the field of dexterous hand hardware. Section 2.2 presents research on grasp planning algorithms. Section 2.3 introduces the work related to datasets and stable grasping. Section 2.4 summarizes the main points of this chapter.

2.1 Dexterous Hand

Currently, there is no precise definition for a dexterous hand. The term "dexterous" is generally understood to mean "Having the ability to perform a difficult action quickly and skillfully with the hands" [Cambridge, 2023]. This explanation emphasizes the importance of hand manipulation skills. Due to the lack of a precise definition, there are no clear constraints on design parameters such as DOF, materials, and actuators. As a result, some end-effectors with fewer DOF can still be considered dexterous hands. A key feature of a dexterous hand can be summarized as one that enables a robot to perform dexterous operations. This feature can serve as a guide for the design of dexterous hands.

Dexterous hand hardware can be classified into different types based on various criteria such as material and actuation method. From the perspective of material, rigid materials can guarantee precise movement while soft materials can adapt to the target more easily. With the trend of combining rigid and soft materials, an increasing number of mixed-material dexterous hands are being developed [Yan et al., 2022]. In this thesis, the main focus is on hands made of rigid materials for more precise movements. Therefore, this section is divided by actuation method and includes two subsections, 2.1.1 under-actuated dexterous hand and 2.1.2 fully-actuated dexterous hand.

2.1.1 Under-Actuated Dexterous Hand

Under-actuated systems have fewer actuators than their DOF. A primary application of this concept is in the design of dexterous hands, which can utilize fewer actuators to control more DOF. Such designs offer several benefits, including reduced cost, less space occupied in the joint, and improved weight distribution. However, a drawback of under-actuated designs is their limited control, which requires reliance on external forces or the environment. Several popular designs in this area are described below.

When designing a dexterous hand, the human hand is often used as a reference due to its remarkable capabilities. The Shadow Dexterous Hand [ShadowRobotCompany, 2013] is a renowned example of a dexterous hand, which uses a tendon-driven actuation method to drive twenty-four DOF similar to the human hand. It has twenty actuators for fully-actuated DOF and four under-actuated DOF. The under-actuated part is similar to the Distal Interphalangeal (DIP) and Proximal Interphalangeal (PIP) joints in the human finger. Although the tendon-driven hand design makes finger movements more flexible, it has the disadvantage of increasing the weight of the arm and the complexity of the tendons inside the hand, and its lifetime is limited. There are two versions of the hand, one with motors and the other with air muscles. The motor version has better control performance and is the latest solution. This dexterous hand is a combination of fully-actuated and under-actuated designs, and more like a fully-actuated hand. It shows the manipulation capabilities of a design that mimics the human hand, and can enable diverse manipulation skills like manipulating a Rubik's cube [Akkaya et al., 2019]. Figure 2.1 shows the Shadow Dexterous Hand and its application in manipulating a Rubik's cube.

Figure 2.1: Shadow Dexterous Hand and application with Rubik's cube

Compared to the traditional materials used in hand design, Three Dimensional (3D) printing offers a very quick design and optimization loop, making it popular for prototype designs. With sufficient precision and an appropriate design, the material itself can be used in the final product. The concept of under-actuated design simplifies the actuation method in anthropomorphic robotic hands, making them easier to use. The OLYMPIC hand [Liow et al., 2019] is a typical hand built using 3D printed

materials. It uses a tendon-driven design, which can achieve a large grasping force with a power grasp. To overcome drawbacks in lifetime and maintenance, this hand takes advantage of a modular design, separating the tendon and actuator into two main parts. The modular design helps to make the tendon-driven mechanism more reliable and the assembly process much easier. The connection design with bevel gear makes the connection more reliable and user-friendly. The separate design can help to keep the actuator with a longer lifetime and easier to replace the parts that are prone to wear. The hand is shown in Figure 2.2.

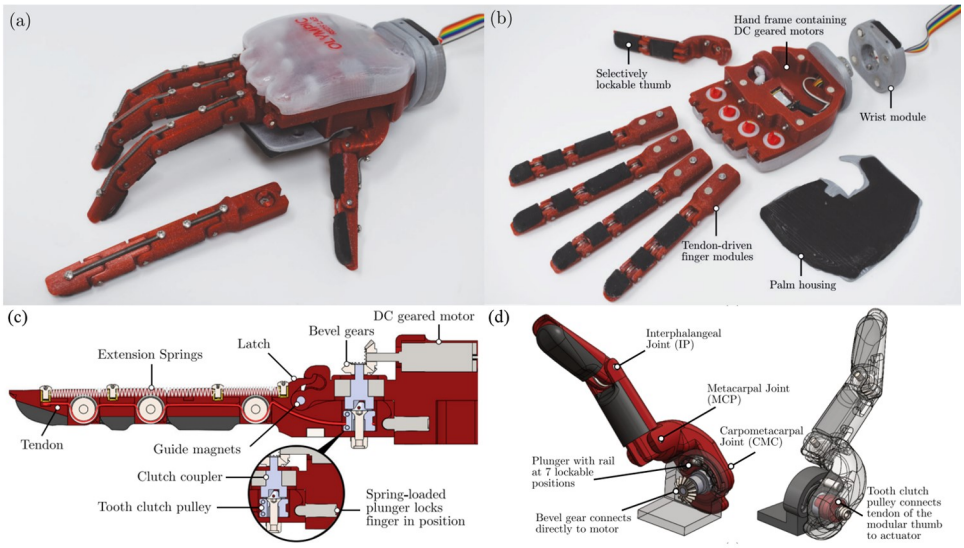
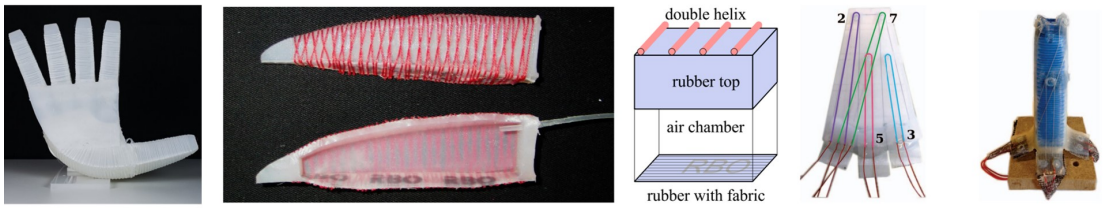


Figure 2.2: OLYMPIC hand and finger structure

As an extension of 3D print material, many soft material hands use 3D structures as molds to manufacture their structure. The RBO Hand 2 [Deimel and Brock, 2016] is a typical example of a hand that uses rubber and compressed air for actuation. The design with soft material and compressed air is common in the soft material hand area and is an efficient way to achieve a power grasp without damaging the target object. The fabric inside the finger modules helps maintain their shape, preventing unexpected deformation. By combining several PneuFlex modules, the anthropomorphic hand can perform thirty-one types of grasp gestures and exhibit dexterity. Combined with a highly stretchable liquid metal strain sensor [Farrow and Correll, 2015], the hand can monitor its deformation status, and assist the grasping task [Wall et al., 2017]. With a 3D printed model, it is easy to produce finger modules with similar designs and materials. However, achieving consistent movement, limited control capability, and preventing air leaks with higher pressure are major issues for such designs. Figure 2.3 shows the hand's structure.

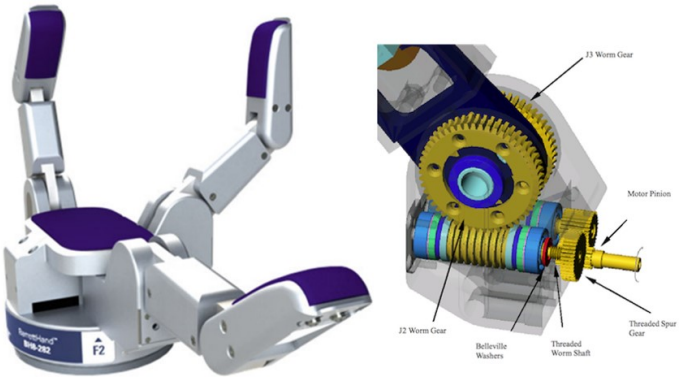
In contrast to anthropomorphic hands with many DOF, some dexterous hands



(a) RBO Hand 2 (b) Cut of a PneuFlex continuum actuator (c) Functional parts (d) Sensor layout (e) Sensorized actuator

Figure 2.3: RBO Hand 2 and liquid metal strain sensor

with different layouts can also provide reliable grasping. The BarrettHand [Townsend, 2000, Barrett, 2012] has been a well-known robotic hand since its introduction in 2000. It is an under-actuated hand with a total of eight DOF and four actuators. Each finger has two DOF, and both joints are driven by a motor through a TorqueSwitch^{TM} mechanism and cable. A clutch and belleville spring washers are used in the mechanism, allowing the front joint angle to change separately when the external torque exceeds a certain value. The two joints in the palm are driven by one gear, which moves in a mirrored direction, making the two finger directions mirrored. The under-actuated design allows the hand to adapt well to the environment, providing sufficient force and adapting to situations when the hand and the environment are not ideally positioned. However, the hand cannot change its finger trajectory without external force. The finger movement follows a fixed trajectory and cannot support fine-tuning of joint values. Thus, the hand can only be used as a parallel gripper until it touches the target, after which the joint motion is determined by the target and environment. The hand and joint structure are shown in Figure 2.4.



BarrettHand TorqueSwitch™ mechanism

Figure 2.4: BarrettHand and TorqueSwitch^{TM} mechanism

The BarrettHand has the advantage of a fingernail, similar to a human nail, which allows it to grasp thin objects such as cards. In terms of sensors, the hand

has tactile sensor modules in its tips and palm, which contain sensor arrays that provide tactile data during grasping. Additionally, fingertip torque can be measured through foil strain gages inside the finger. Figure 2.5 shows the fingernail and sensor designs.

BarrettHand fingernail and tactile sensing array Fingertip torque sensor

Figure 2.5: BarrettHand fingernail and sensors

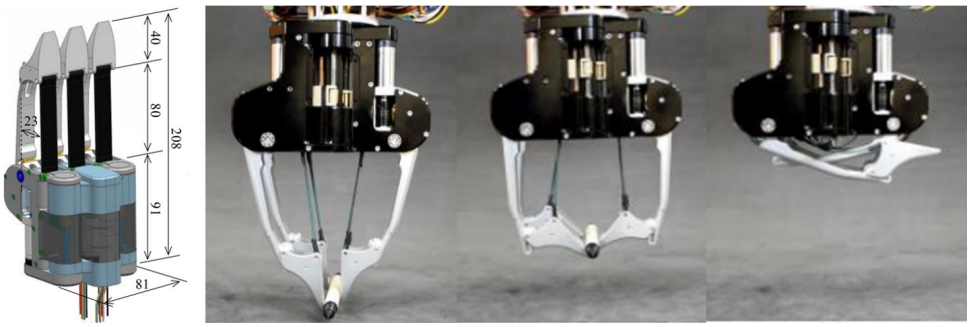


Figure 2.6: BLT Gripper and compliant grasping process

In addition to hands made of a single material, there are also designs that combine both rigid and soft materials. The BLT Gripper [Kim et al., 2020] is one such design that takes advantage of the benefits of both materials. With five degrees of freedom for three-finger hand, each finger module contains one motor providing rotational motion, one in the palm providing three fingertip motions through a lead screw, and one for finger base rotation. The combination of finger joint motion and fingertip motion allows the hand to perform precise pinches and compliant grasps with ease. The grasping process of the BLT gripper can be broken down into several steps, including parallel pinching, inclining the fingertip, and compliant grasping using a belt. The belt provides good compliance and enhances the stability of the grasp. It is a novel material that is different from soft materials like rubber. Kinematic analysis in the design is critical for achieving optimal dexterity. The hand also features a modular design and uses motors and gearboxes to transmit power directly. Figure

2.6 shows the hand's design.

The hands introduced are under-actuated dexterous hands that with fewer actuators than their DOF. Through various driver methods, materials, actuators, and combinations, these hands have different advantages in their respective areas. Characteristics of these dexterous hands can include high DOF, modular fingers, soft materials, unique layouts, sensing capabilities, combined materials, and more. These features can aid in the design of the dexterous hand in this thesis.

2.1.2 Fully-Actuated Dexterous Hand

A fully-actuated dexterous hand has the same number of actuators as its DOF, allowing for precise control and greater dexterity. However, this design also has limitations, such as output torque, volume, and weight, which must be balanced to achieve optimal performance. Here are some examples of typical fully-actuated dexterous hands.

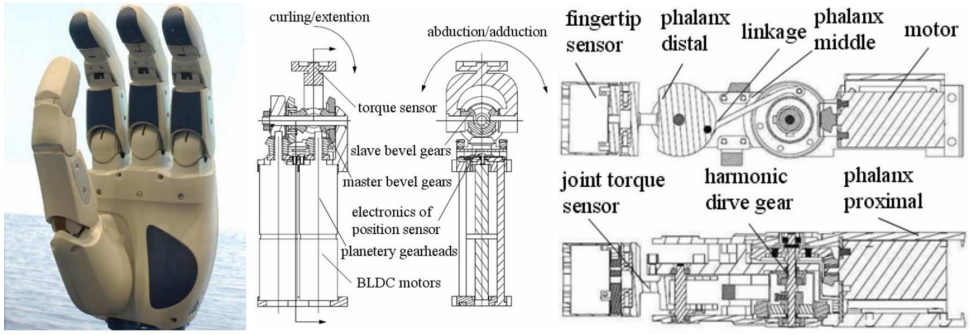


Figure 2.7: DLR-HIT-Hand and finger joint structure

Like the under-actuated hand mimics the human hand, the fully-actuated hand can also mimic the human hand. The DLR-HIT-Hand [Liu et al., 2008] is a fully-actuated hand that mimics the human hand with a similar layout. It has thirteen DOF, with three fingers and one thumb in total. Unlike the tendon structure used in the Shadow Dexterous Hand, the DLR-HIT-Hand uses a BrushLess Direct Current (BLDC) motor to drive the joint through a harmonic reducer. The connection between the DIP and PIP joints is mechanically coupled by a rigid linkage and driven by one motor. The finger base uses two motors to control the two DOF separately. All of these actuators are mounted inside the finger and palm, which makes the total volume smaller. However, the joint output performance is limited by the small motor inside the joint with a value of 10 N fingertip force. The motor and reducer inside the joint may be more fragile due to the limited torque and rigid driven structure. The control board is located inside the finger module, allowing for faster data process and motion

control. The main structures of the hand are shown in Figure 2.7.

Due to the complex structure of the tiny joint, maintenance can be difficult inside one joint. Similar to the OLYMPIC hand, this issue can be addressed through the use of a modular finger design that separates the fragile and strong components. This approach has also been implemented in other fully-actuated hand designs. The Sandia Hand [Quigley et al., 2014] is a dexterous hand with a real modular finger design, where each finger module can be easily replaced. There are three DOF in each finger module, including two for extension and flexion and one for abduction and adduction. This design makes it easier to replace the finger module, which benefits maintenance. The layout and number of finger modules can also be changed with different palm designs according to different requirements. The hand is shown in Figure 2.8.

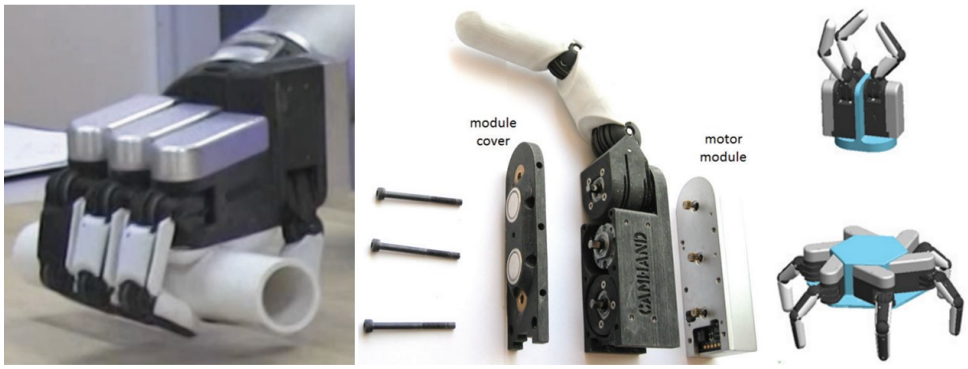
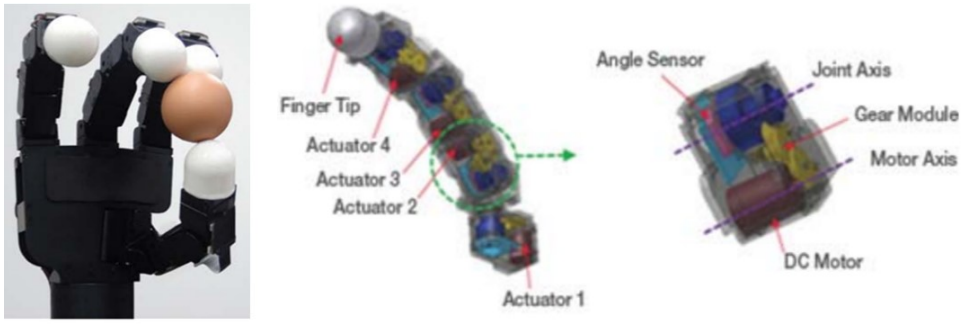


Figure 2.8: Sandia Hand, finger module, and various extensions



Allergo Hand Finger module DC motor unit

Figure 2.9: Allergo Hand, finger module, and Direct Current (DC) motor unit

In addition to the modular finger design, the modular concept can also be applied at the joint level. The Allergo Hand [ROBOTICS, 2012] is a typical design with modular motor units, where each finger has four DOF. The actuator in each joint uses the same DC motor unit, making it easier and more cost-effective to maintain the hand. The speed of each finger joint is fast enough for most daily life scenarios, including

dynamic grasping flying objects [Salehian et al., 2016]. However, one drawback of this design is that the motor in the base joint bears several times more torque than the motor at the front tip, requiring a balance between total dimensions and output force. Additionally, this hand needs to add external tactile sensors separately for manipulation tasks that require tactile feedback. The hand, finger, and DC motor unit are shown in Figure 2.9.

When designing a dexterous hand with high dexterity and no limitations in shape and layout, many specialized designs can be considered. One such example is the roller grasper [Yuan et al., 2020], which has a unique structure. To achieve its dexterity, the hand combines complex hand actions into two types of motion: grasping and rolling. By combining rollers and fingers, the hand can perform certain tasks more easily than a human hand, and in some cases, the rotation of objects can be controlled more effectively. This type of end-effector has the potential to improve robot manipulation efficiency by extracting key features in the target scene and using a specific design to cover the required function. It is a good example of balancing versatility and efficiency. Figure 2.10 shows the design of the hand.

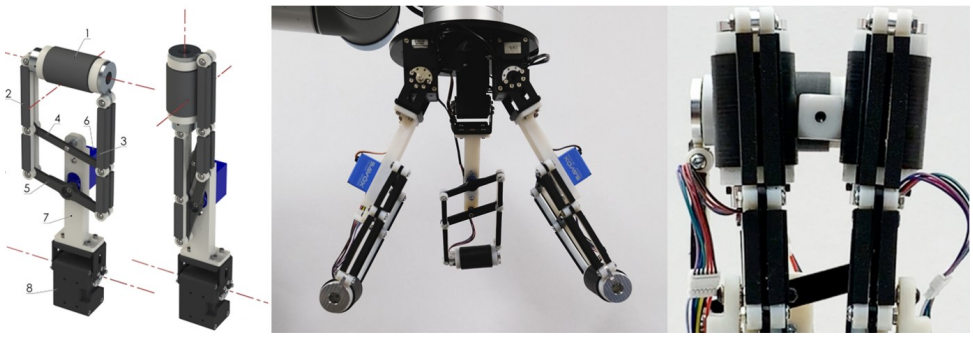


Figure 2.10: Roller grasper module and example of rotating small cube

The fully-actuated dexterous hands introduced in this discussion demonstrate the potential benefits of precise control over the DOF and enhanced dexterity. The use of DC motors in many designs enables a more compact layout, while the design of modular fingers and modular motor units simplify the design and maintenance process and promotes versatility across different applications. With the diverse manipulation functions provided by the fully-actuated design, a practical dexterous hand design can consider a wider range of possibilities. Then, the dexterous hand in this thesis uses a fully-actuated design for more practical functions.

2.2 Grasp Planning

Grasp planning for a robot involves determining how to grasp an object, including the pose of the arm and the parameters of the end-effector. When using a dexterous hand with a higher DOF, the process becomes more complex. Grasp planning methodologies can be broadly divided into two approaches: analytic and data-driven [Bohg et al., 2013]. The analytic approach uses mathematical models to generate feasible grasps based on the physical properties of the object, while the data-driven approach relies on learning from data to generate grasp plans. In this section, these two approaches and related datasets are introduced in more detail in three subsections, 2.2.1 analytic approach, 2.2.2 data-driven approach, and 2.2.3 dataset for grasp planning.

2.2.1 Analytic Approach

Analytic approaches have been developed over a long period of time and the field of grasp planning was dominated by them before 2000 [Zhang et al., 2022]. At the core of the analytic approach is grasp closure, which can be categorized into form-closure and force-closure. If a contact is self-sufficient in maintaining itself without reference to the applied forces, it can be regarded as form-closure. If a contact is maintained by the action of certain forces, it can be regarded as force-closure [Dizioğlu and Lakshiminarayana, 1984]. The differences between these two typical cases can be easily observed in Figure 2.11.

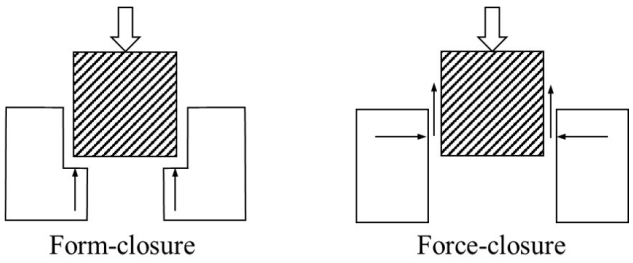


Figure 2.11: Typical cases of form-closure and force-closure

The grasp-closure problem and its assumptions have been extensively studied in the literature. Various works have investigated the verification of force-closure grasps for both Two Dimensional (2D) and 3D objects by making certain assumptions about the contact model and contact point location [Zhang et al., 2022]. In Liu's work, a 2D object grasping with force-closure can be simplified to a nonmarginal equilibrium with a three-finger hand, and the 3D case can be derived from the 2D case [Li et al., 2003]. Furthermore, it can also verify that a 3D object can be grasped with force-closure by a three-finger hand. By applying different methods to verify grasp-

closure conditions, a grasp plan that satisfies the requirement of grasp-closure can be obtained.

In addition to ensuring grasp-closure, it is also important to select the best grasp plan during the grasp planning process. A widely used solution evaluates the grasp quality using specific criteria based on the Grasp Wrench Space (GWS). The GWS is defined as the convex hull of all possible wrenches that could be imposed through the contact points [Ferrari and Canny, 1992]. The wrench is a vector W defined as shown in Equation 2.1, where F is the force component and T is the torque component. The quality of a grasp can be defined in terms of the GWS and evaluated through convex hull calculation.

$$W = \begin{bmatrix} F \\ T \end{bmatrix} \tag{2.1}$$

Based on the GWS, Li proposed a method for evaluating grasp quality and generating a grasp plan. This method uses the finger position on the touched edges to obtain the Graspable Finger Position Regions (GFPR), which represents all stable grasping solutions within the GWS [Li et al., 2002]. The GFPR can be obtained by computing the hyperplane of each contact point. By considering the errors and adjustment range of each grasp plan, solutions inside the inscribed hypersphere of the GFPR provide a more stable status. The diameter of the hypersphere can be used as a grasp quality criterion. The finger position value at the center of the hypersphere is considered the best grasp plan. An example of the GFPR for three fingers is shown in Figure 2.12. Overall, Li's method provides a useful approach for selecting the optimal grasp plan by considering both grasp stability and quality, based on the GWS and GFPR.

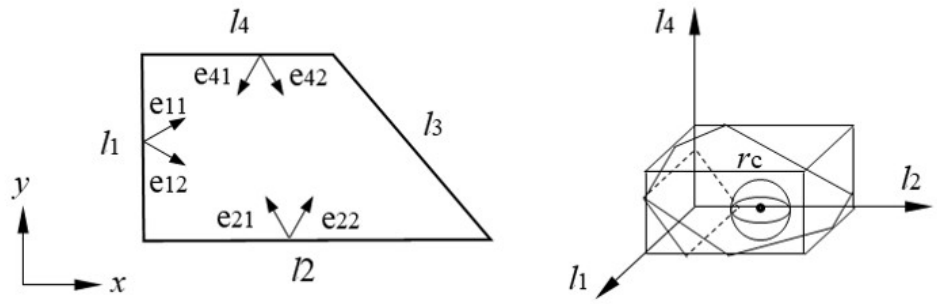


Figure 2.12: The hypersphere in GFPR

There are many similar analytic solutions for different situations, as shown in Table 2.1 [Zhang et al., 2022]. However, the application of these solutions is limited by various assumptions. The representation of the grasp plan includes two

main methods: Contact points and Independent Contact Regions (ICRs). The complex nature of real-world objects and target scenes makes the calculation more challenging, especially for 3D objects. To address more complex situations, 3D simulators such as GraspIt! [Miller and Allen, 2004], V-REP [Rohmer et al., 2013], Gazebo [Koenig and Howard, 2004], and Pybullet [Pybullet, 2016] can be useful tools. GraspIt! is a widely used simulator that focuses on the grasping function and can easily generate grasp candidates that meet grasp-closure requirements through simulation. With the help of these simulators, grasp planning can become easier, and there remains a gap between reality and the simulation environment.

Table 2.1: Summary of some analytic grasp planning methods

Author & Year	Representation	Type	Fingers	Object
[Nguyen, 1988]	Contact points & ICRs	Frictional	2,3,4,7	Polygons & Polyhedra
[Markenscoff and Papadimitriou, 1989]	Contact points	Frictionless	3,4	Polyhedra
[Faverjon and Ponce, 1991]	Contact points	Frictional	2	Curved Shapes
[Ferrari and Canny, 1992]	Contact points	Frictional	2,3	Polygons
[Ponce and Faverjon, 1995]	ICRs	Frictional	3	Polygons
[Ponce et al., 1997]	ICRs	Frictional	4	Polyhedra
[Smith et al., 1999]	Contact points	Frictional	2	Polygons
[Liu, 2000]	Contact points	Frictional	n	Polygons
[Ding et al., 2001]	Contact points	Frictional	n	3D Objects
[Zhu and Wang, 2003]	Contact points	Frictional	n	3D Curved Objects
[Pollard, 2004]	Contact points & ICRs	Frictional	n	3D Objects
[Jia, 2004]	Contact points	Frictional	2	Curved Objects
[Cornelia and Suárez, 2005]	ICRs	Frictionless	4	2D Discrete Objects
[Cornella and Suárez, 2005]	ICRs	Frictional	n	Polygons
[Niparnan and Sudsang, 2006]	Contact points	Frictional	3	2D Objects
[Roa and Suárez, 2008]	ICRs	Frictional	n	3D Objects
[Roa and Suárez, 2009]	ICRs	Frictional	n	3D Objects

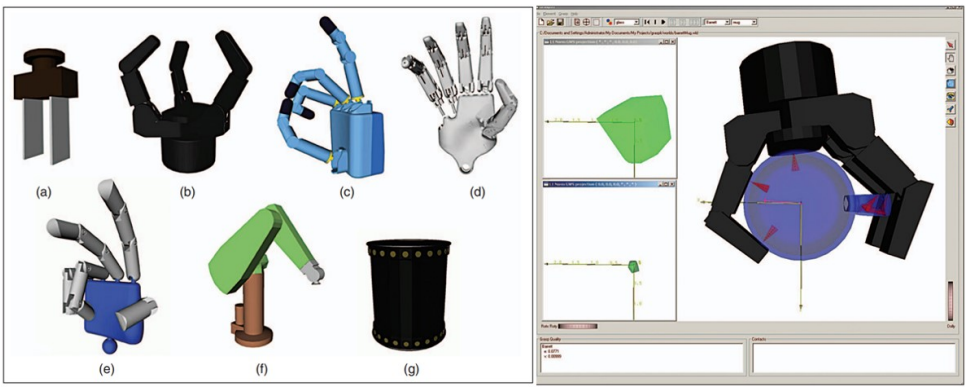


Figure 2.13: End-effectors and grasp poses in GraspIt!

In conclusion, the analytic solution for grasp planning can provide an optimal grasp in theory and is highly dependent on the input physical parameters such as the hand, object, and physical characteristics. It provides a solid theoretical foundation for grasp planning. However, this approach may not be adaptable to some complex environments, especially for some unsatisfied assumptions. As a supplement, data-driven approaches have been developed to address some limitations and provide adaptive grasp planning solutions.

2.2.2 Data-Driven Approach

In recent years, with the rise of deep learning, data-driven approaches for grasp planning have gained popularity among researchers [Bohg et al., 2013]. These methods often use neural networks to perform grasp planning and evaluate grasp quality, resulting in a stable grasp plan. The following content discusses some typical solutions based on data-driven approaches.

With some modifications, the bounding box of an object in an object detection application can be modified to represent a grasp plan. Guo trained a Deep Neural Network (DNN) using 2D images to enable a robot to grasp a target object [Guo et al., 2017]. This approach is mainly based on object detection neural networks, where the grasp rectangle is rotated using the rotation angle parameter, and the output parameters include x, y, width, height, and angle. It provides an efficient solution for generating grasp plans for different objects. A typical representation of a 2D grasp plan in DNN work can be seen in Figure 2.14 [Guo et al., 2017].

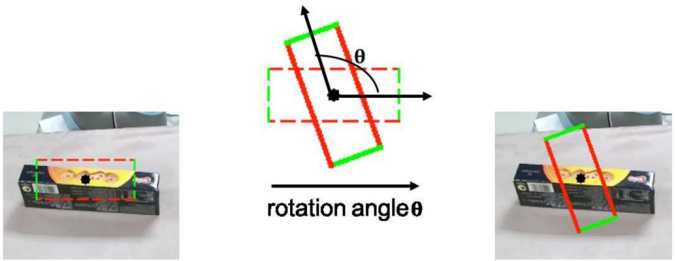


Figure 2.14: Definition of 2D grasp plan

To gather more data to train the network, Pinto addressed this issue by using the Baxter robot to generate 50,000 data points through self-supervision [Pinto and Gupta, 2016]. The grasping area was set on a table with a Table-Top scene, and the vision sensor provided real-world grasp feedback. This approach involves labeling the grasp poses in 2D images, validating them through grasp trails, and gathering data over a long period. With enough data, good results can be achieved. The experimental scene from this study is shown in Figure 2.15.

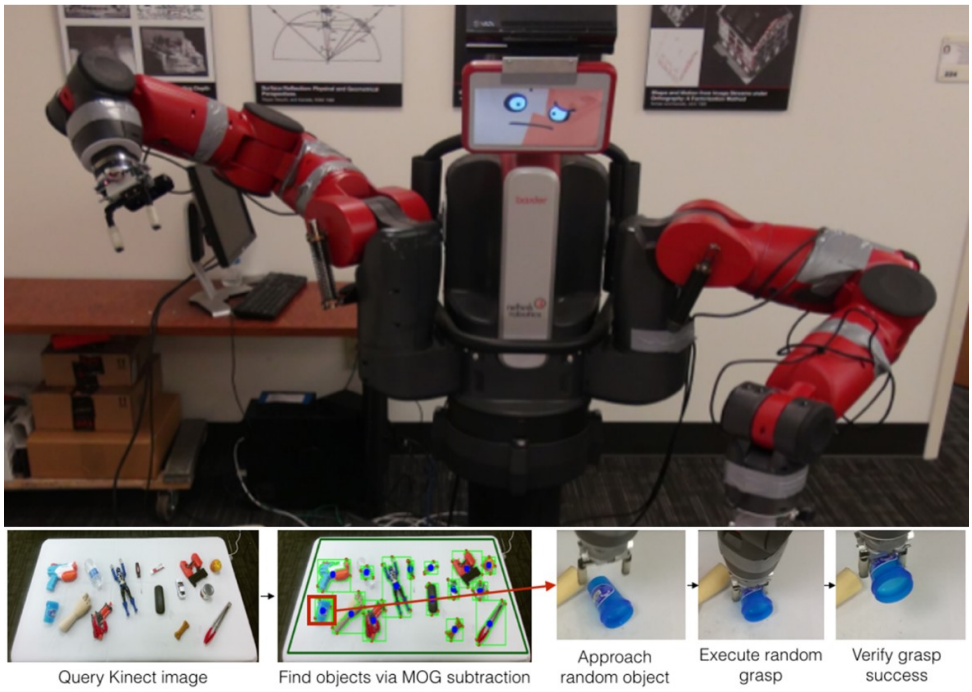


Figure 2.15: Data collection with Baxter robot

Using multiple robots can improve the efficiency of the data collection and training process compared to using a single robot. Levine used 14 robots to perform random grasps over 800,000 times, collecting grasp data for training a DNN [Levine et al., 2018]. These robots were set up with a uniform layout and followed a consistent data collection procedure, grasping items within a designed bin to ensure a continuous loop of grasping. This approach shows promise in terms of data gathering and sharing, as the robots can share the data they collect and improve their capability as a collective. The test scene is shown in Figure 2.16.

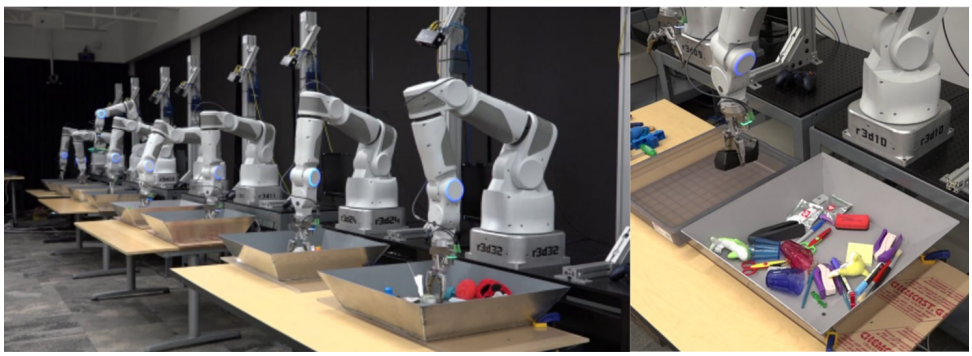


Figure 2.16: Collecting grasping data with fourteen robots

To improve the performance of 2D grasp planning with a neural network, apart from providing more training data, segmenting the target object first can

help to simplify the grasp planning issue. Dong proposed a two-step approach [Dong et al., 2021]. First, the target object is segmented using Mask R-CNN [He et al., 2017]. Then, the segmented object is input into the grasp planning network. This design simplifies the background of the target object and improves performance compared to other 2D grasp planning models. This solution highlights the importance of segmentation in the environment for improving grasp planning performance and can also help simplify the generation of training samples for the grasp planning network. The image data processing procedure is shown in Figure 2.17.

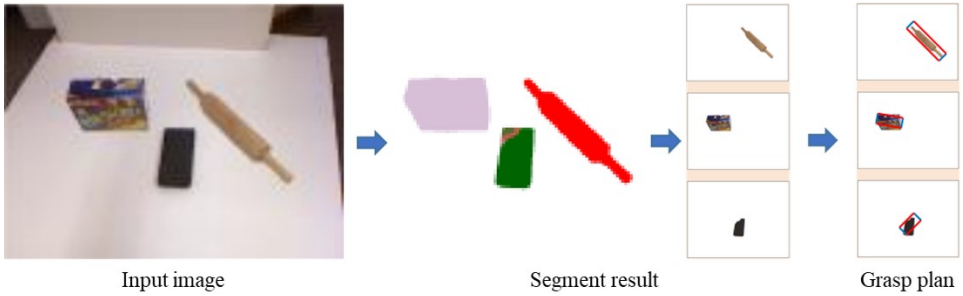


Figure 2.17: Grasp plan procedure of 2D grasp planning with Mask R-CNN

Since 2D images cannot provide comprehensive data in real-world scenarios, using 3D data to train neural networks can improve performance. Mahler set up a simulation environment and built a dataset consisting of millions of point cloud data, which was used to train a Grasp Quality Convolutional Neural Network (GQ-CNN) with an analytic metric. The GQ-CNN selects the best grasp plan, achieving a 93% success rate with eight types of known objects[Mahler et al., 2016, Mahler et al., 2017, Mahler et al., 2018]. This work has been extended to the parallel gripper and suction cup gripper. With a large dataset, this approach allows for easier application of different end-effectors. The experiment setup and process are shown in Figure 2.18.

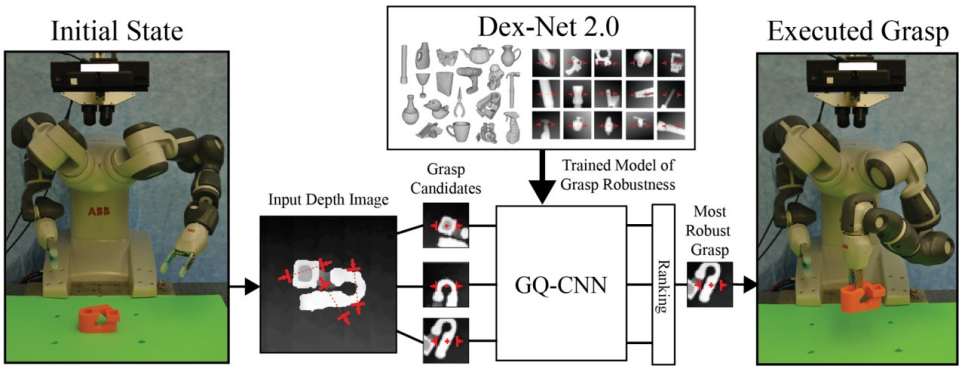


Figure 2.18: Grasp plan procedure of Dex-Net 2.0

When it comes to grasping with diverse end-effectors in robots, it is crucial to have a method that is universally applicable across different end-effectors. AdaGrasp is an approach that addresses this challenge by training the network with twelve different end-effectors, enabling it to generate various grasp plans based on the specific characteristics of each end-effector [Xu et al., 2021]. This approach represents a step towards creating a universal network for different end-effectors. The evaluation of grasps in AdaGrasp is performed using a DNN, which is trained with the outcome of success or failure. Although the target scene and end-effector types in this method are limited, the approach holds promise for the development of universal grasping networks.

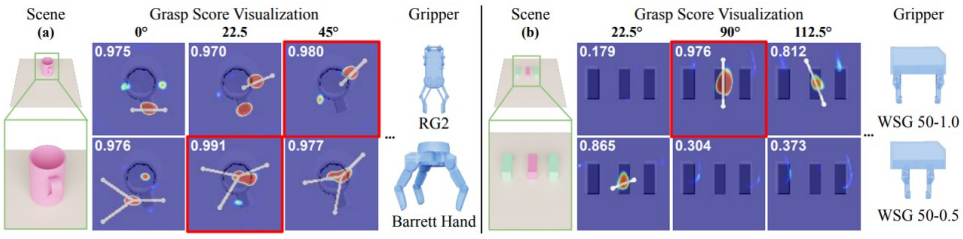


Figure 2.19: Adagrasp with different end-effectors

Considering these introductions, it is clear that data-driven approaches are data-intensive, and many existing solutions offer efficiency, adaptability, and strong performance. However, certain failure cases still exist, and the quality evaluation part primarily relies on data training. To improve the precision of grasp planning and accelerate the learning rate of the grasping process, a combination of analytic and data-driven methods can be used. This combination can help overcome the limitations of data-driven solutions by incorporating prior knowledge and expertise into the grasp planning process. The following subsection introduces datasets that are related to the prior knowledge of grasp planning.

2.2.3 Dateset for Grasping Planning

Datasets play a crucial role in data-driven approaches, serving as a common foundation for evaluating different solutions based on standardized criteria. Simulation-based datasets are widely used for data collection due to their convenience and ability to encompass diverse objects and scenarios. For example, Goldfeder built the Columbia dataset using the simulation software GraspIt! [Miller and Allen, 2004], which contains nearly 22,000 grasping samples involving daily life items [Goldfeder et al., 2009]. Some samples from this dataset are showcased in Figure 2.20. These simulation-based datasets provide a good starting point for grasping and can help researchers optimize their solutions with fewer trials in

the real world.

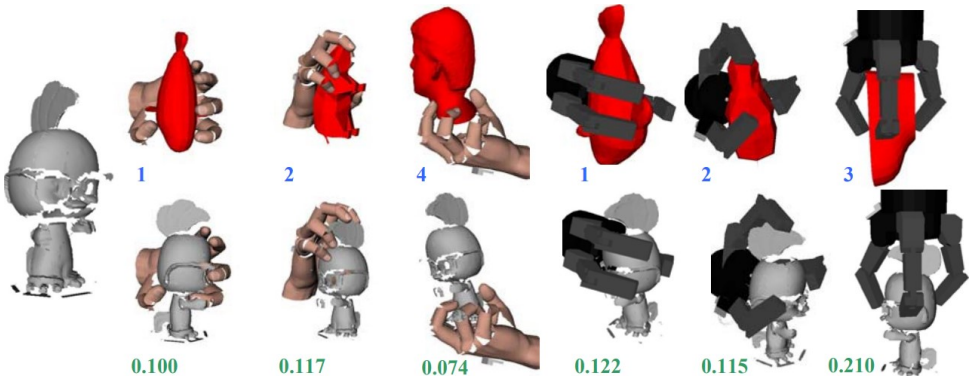


Figure 2.20: Grasp pose data in Columbia dataset

Physical world benchmarks are also crucial for evaluating solution performance. Various works have defined some object sets for research in the grasping domain. Object sets such as the Karlsruhe Institute of Technology (KIT) object set and the Yale-CMU-Berkeley (YCB) object set provide common objects along with corresponding 3D models, enabling researchers to focus on algorithms and ensuring more universal output [Kasper et al., 2012, Calli et al., 2015]. Some objects from the YCB object set are displayed in Figure 2.21.

Figure 2.21: Objects in YCB object set

To gather robot grasping data in the physical world and make it easier for robots, Zhang generated demonstrated data through Virtual Reality (VR) and applied it to object manipulation [Zhang et al., 2018]. This process involves mapping human demonstrated actions to robot joint actions. The training results are significantly influenced by the number of samples, with the success rate improving from 20% to 80% when the sample number is increased from 11 to 109. These results highlight the importance of high-quality datasets for robotic manipulation. The experiment scene is shown in Figure 2.22.

Compared to data produced by simulators or robots, there is a vast amount of video data related to human manipulation available on the internet. Utilizing such data can be the most convenient way to gather data. Pierre conducted a study in this

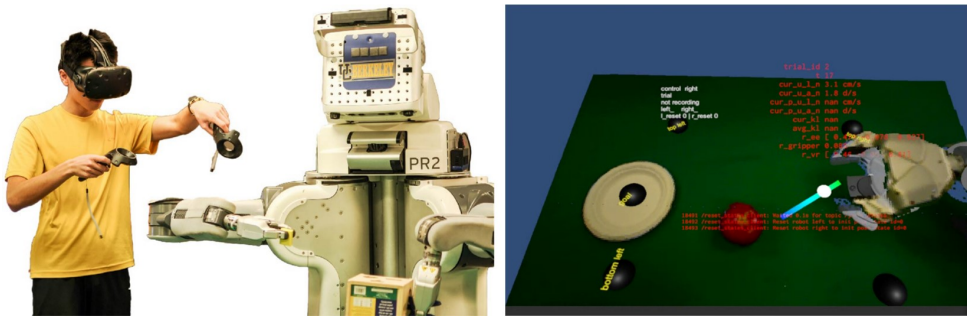


Figure 2.22: Gather data with PR2 through VR control

direction, utilizing human manipulation video data to teach robots how to perform the pouring tasks, as illustrated in Figure 2.23. Leveraging this data helps acquire a larger dataset without the constraints of robotic hardware [Sermanet et al., 2018]. However, this approach requires additional efforts in key-point detection, knowledge transfer from the human body to the robot, and related technologies. The value of this type of dataset will increase as the extraction of valuable data from videos becomes more precise and convenient.

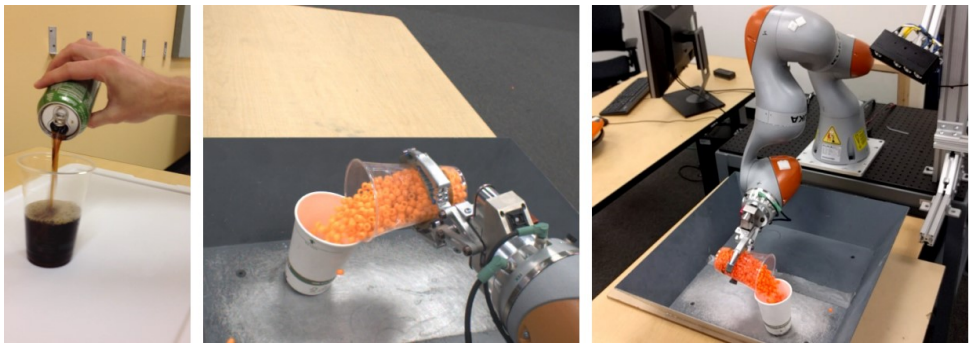


Figure 2.23: Learning pouring action from human action

Based on the aforementioned datasets, virtual datasets accelerate algorithm development iterations, object sets provide the common objects, robot datasets provide convenient data for robotic systems, and human datasets hold promise for future advancements. The importance of the dataset for data-driven approaches is clear. The present thesis work is also inspired by these works and utilizes data to develop deep learning-based grasp planning.

2.3 Stable Grasping

Stable grasping focuses on the procedures that occur after pre-grasp manipulation, including grasp acquisition and post-grasp manipulation. Compared to grasp plan-

ning in a static situation, stable grasping mainly focuses on the dynamic process. The key to stable grasping is evaluating grasp stability. In order to achieve stable grasping under various scenarios, a large amount of data is required to identify unstable conditions. This section introduces the related data and algorithms, and is divided into two subsections: 2.3.1 dataset for stable grasping, and 2.3.2 grasp stability evaluation.

2.3.1 Dataset for Stable Grasping

Similar to human senses, both visual and tactile data are important for robots to achieve better performance. However, in the case of unstable grasping, visual data may be occluded by the hand, making tactile sensing more important for evaluating grasping stability. Since tactile data needs to be acquired through contact between the object and sensor, this collection process is more complex than collecting visual data.

To evaluate grasp stability, the interaction between the hand and object is crucial. Slip is a key feature of unstable grasping and serves as an indicator of instability. As a result, slip detection during the grasping process has been a focal point for many researchers [Heyneman and Cutkosky, 2016]. To acquire pure slippage data, experiments can be conducted with relative motion between sensors and objects. The experiment involves moving the finger to generate slippage data, and an example is shown in Figure 2.24 [Kobayashi et al., 2012]. These data can be used for slip detection and stability evaluation.

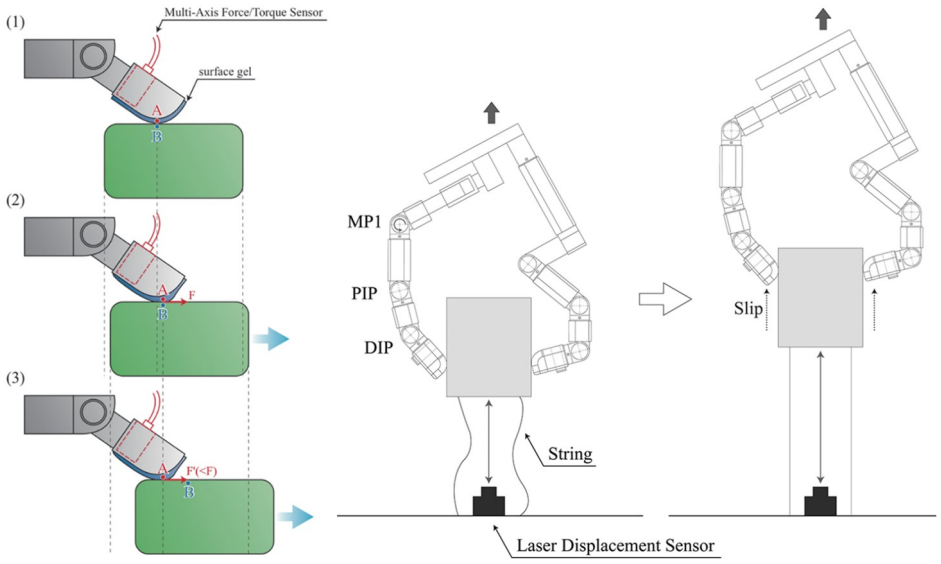


Figure 2.24: Gather slip data with relative motion and verify the slip detection

To achieve stable grasping with learning, it is important to have both pos-

itive and negative data in tactile datasets. These data can be used to identify features of instability and develop algorithms for grasp stability evaluation [Stachowsky et al., 2016, Liu et al., 2018, Hildebrandt et al., 2008]. The occurrence of unstable cases during normal grasping can provide diverse data. Some researchers have acquired data during normal grasping cases. Chebotar used the BioTac sensor to gather data during random grasping processes, providing more diverse data with a 46% failure rate in 1,000 grasps [Chebotar et al., 2016a, Chebotar et al., 2016b]. The test scene was set inside a bowl to improve experiment efficiency. This type of data includes more than just slip and can be used for comprehensive grasp stability evaluation. The experiment scene is more useful in Table-Top grasping works and can be seen in Figure 2.25.

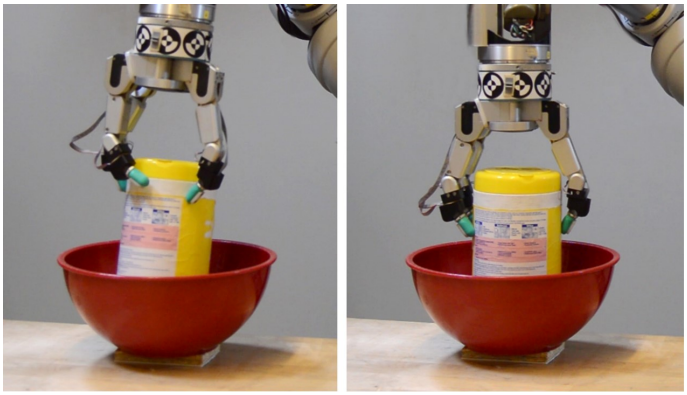


Figure 2.25: Experimental setup for collecting tactile data during grasping

In contrast to the piezoresistive mechanism used in BioTac, there are other sensors that based on visual sensors. Li used the GelSight sensor to acquire grasp data and build a dataset with GelSight and external visual data [Li et al., 2018]. This type of sensor provides tactile data in the form of visual images, which have the same format as external visual data. Both types of images are input into a DNN model to realize the slip detection function. The test scene and data can be seen in Figure 2.26.

The introduction of datasets for stable grasping shows the differences between gathering data from designed relative motion and real grasping processes. While current datasets primarily focus on tactile data, new datasets are emerging that encompass the entire grasping process. A comprehensive dataset that combines various types of data, including tactile and visual information, holds the potential for enhancing fine manipulation skills and advancing robot capabilities. Such a dataset would enable researchers to reconstruct the entire process and conduct extensive simulations, accelerating research progress.

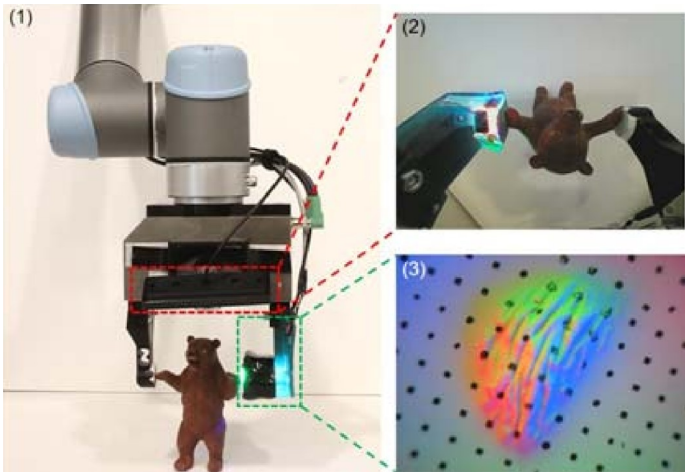


Figure 2.26: Experiment with GelSight, external image, and GelSight image

2.3.2 Grasping Stability Evaluation

Grasp stability is used to describe the grasp status, and its evaluation can indicate the grasp status. For an unstable grasp, there may be features such as vibration or slip. These features can be used for evaluating grasp stability. Considering that external conditions may change during a manipulation task, grasp stability prediction should be used in the loop control of stable grasping.

Predicting slips before gross motion occurs can ensure more stable grasping. This predictable incipient slip event can be evaluated using measurements of changes in contact position, force, and vibrations [Stachowsky et al., 2016]. For force data, some works use a threshold to judge the status in the time and frequency domain. Such solutions can achieve good results under specific situations but require experience in adjusting the threshold. Figure 2.27 shows slip detection with a threshold on wavelet coefficients [Zhang et al., 2016].

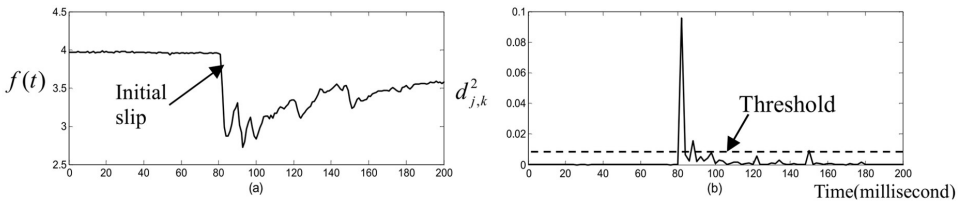


Figure 2.27: Judge status through a threshold on frequency domain

To model grasp stability prediction with an optimal method, it is important to use a high quality sensor. Since the grasp is held by the force status, a force sensor is a direct way to sense the force change. This allows for immediate prediction of instability based on the changed value. Barrett built a device to gather the slip data, compared

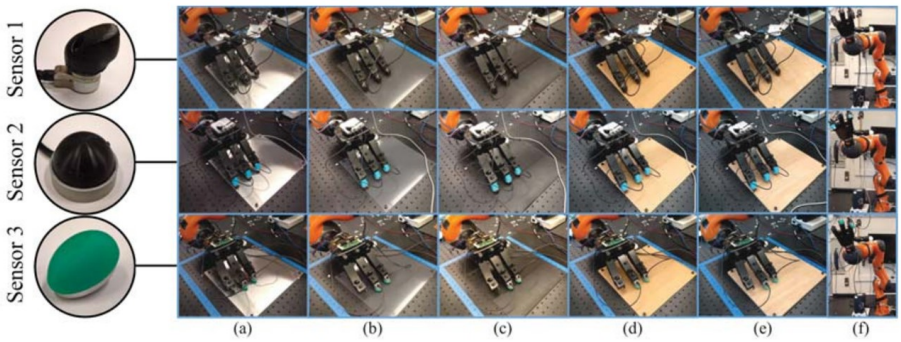


Figure 2.28: Acquiring slip data with different tactile sensors

data from three different sensors, and treated the issue as a classification issue in the frequency domain [Heyneman and Cutkosky, 2016]. Karl used three kinds of tactile sensors to gather the data and classified the status through Long Short-Term Memory (LSTM) with data in the time domain [Van Wyk and Falco, 2018]. The experiment process is shown in Figure 2.28.

Compared to the force sensor with faster change, visual-based sensors judge stability through deformation changes. These changes can only be detected after a large enough force value changes [Dong et al., 2017]. This makes the visual sensors not the optimal choice, but they can provide some other features related to deformation and texture. Figure 2.29 shows the prediction through a visual-based tactile sensor.

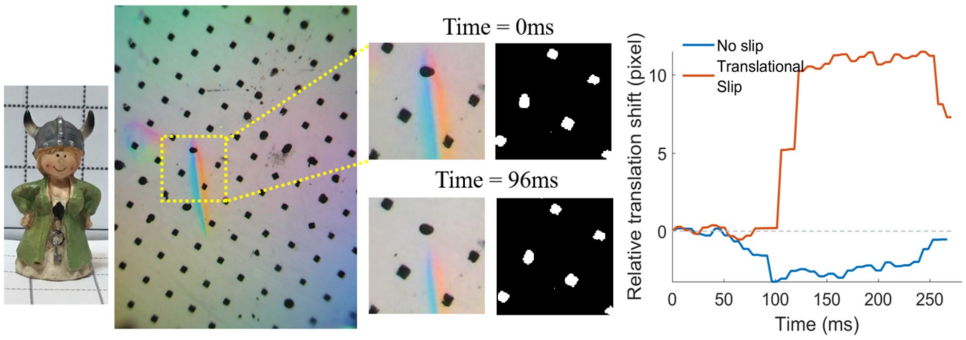


Figure 2.29: Relative translation shift on the visual sensor

By using data from a force sensor and setting a threshold to judge grasp stability based on time and frequency domains, reasonable prediction results can be obtained for certain tested items. There are many methods that treat stability issue as a classification issue with good results. However, the object's material, weight, and shape may influence the classification results. Therefore, a large library is required to maintain these features [Cavallo et al., 2014, Fernandez et al., 2014]. Like the application of LSTM, a learning-based approach with generalization capability is more suitable for handling complex situations.

2.4 Conclusion

This chapter introduces the state of the art in three related directions of this thesis. The current progress is constantly being updated, and trends in different areas can be clearly seen in the latest research. The development of the dexterous hand is moving in a more practical direction. Influenced by the development of robotic arms and mobile bases, the usage of the dexterous hand is closer to real-world applications. The trend in grasp planning is more clearly shown, with analytic approaches dominating research before 2000 and data-driven approaches becoming more popular in recent years. Combining analytic and data-driven methods is becoming a clearer trend and can result in more efficient outcomes. The area of stable grasping is not as popular currently, limited by hardware and unresolved grasp planning issues. However, there have been some breakthroughs in grasp stability due to datasets and learning methods. More extensive datasets and breakthroughs in learning methods are still necessary. With these trends in mind, the work in this thesis is introduced in the following chapters.

Part II

Dexterous Hand

Chapter 3

Dexterous Hand Design

A dexterous hand is the fundamental hardware in this thesis. With a well-designed dexterous hand, robots can perform tasks more efficiently. As introduced in the first chapter, the goal of this thesis is to design a user-friendly and practical modular dexterous hand for robotic manipulation tasks and serve as practical robotic hardware for robotic applications. Section 3.1 presents the design concept of the dexterous hand. Section 3.2 introduces the detailed design of the key components. Section 3.3 presents the dexterous hand from the mechanical, electronic, and embedded perspectives. Section 3.4 provides a summary of this chapter.

3.1 Design Concept

For stable grasping and manipulation, a dexterous hand with at least three fingers is necessary. As requirements become more diverse, additional fingers can enhance dexterity. However, it's crucial to balance complexity and performance in hardware design, which is the core concept of the design.

Due to the advantage of modular design, the necessary fingers can be equipped as needed, which can be adapted to different situations and reduce hardware costs. This hand employs a modular design and chooses a three-finger dexterous hand as the basic version. To achieve a stable grasping function, tactile sensing capability is necessary and a key feature of a dexterous hand. These two features are key points in the design of a dexterous hand and are introduced in two subsections: 3.1.1 degrees of freedom and hand layout, and 3.1.2 sensing capability.

3.1.1 Degrees of Freedom and Hand Layout

In a modular design, the DOF and hand layout are important factors. These two elements determine the drive method, motion range and manipulation capability.

Detailed discussions on these topics can be found in follows: 3.1.1.1 finger degrees of freedom, and 3.1.1.2 hand layout.

3.1.1.1 Finger Degrees of Freedom

As previously mentioned in the introduction subsection 1.1.1, the human hand has a total of twenty-one DOF: five DOF in the thumb and four DOF in each finger. Since the thumb has a unique base and location, the finger module design for the dexterous hand primarily refers to the other four fingers. The design related to the thumb is discussed in the 3.1.1.2 hand layout.

Among the four DOF shown in the finger structure diagram in Figure 3.1, the DIP joint is typically an under-actuated joint and cannot be fully controlled by most people. This adaptive structure can be realized with elastic or soft material. Considering function and cost, a finger module with fully-actuated DOF can realize more diverse functions and the adaptive structure can be an extension of this hand. Therefore, the DIP DOF is ignored in the basic design, and each finger has three DOF: two for extension and flexion, and one for abduction and adduction, all of which are fully-actuated to mimic the function of the PIP joint and Metacarpophalangeal (MCP) joint.

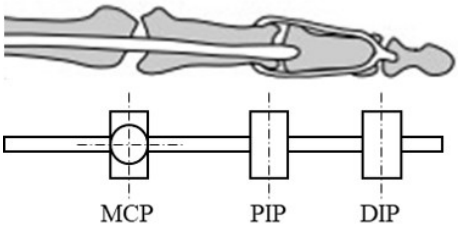


Figure 3.1: DIP, PIP, and MCP in one finger

With a total of three DOF, it is critical to assign these degrees of freedom to one finger. Compared to the MCP joint, integrating two DOF into the mechanical structure requires a structure like a ball hinge or universal joint, which makes the drive structure more complex, limits actuator selection, and reduces the structural strength. Considering design complexity, total weight, and output torque, it is better to separate these two DOF, assigning one DOF to the finger and another to the palm. The number of DOF in one finger module is decided to be two, and the other DOF is assigned in the palm. The joint change can be seen in Figure 3.2.

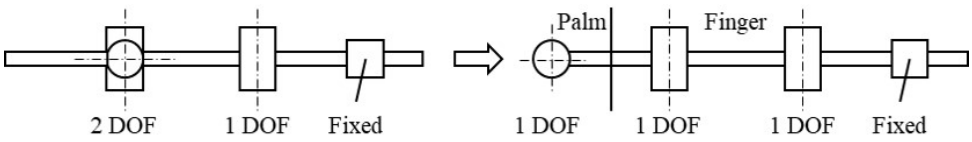


Figure 3.2: The finger DOF layout change

With the number of DOF, the modular concept needs to be defined at a specific level. There are two different choices: driver module, and component module. The AllergoHand is a driver module design [ROBOTICS, 2012], while the SandiHand is a component module design [Quigley et al., 2014]. The driver module has limitations in joint torque but has a simple design, and the cost of the actuator can be lower. The component module has a complex design but offers flexible choices in components. Referring to these factors, a comparison table can be seen in Table 3.1.

Table 3.1: Comparison of two module types

Type	Module volume	Total volume	Structure complex	Module assembly	Torque	Cost
Driver module	Small 2	Big 1	Simple 2	Hard 1	Low 1	Low 2
Component module	Big 1	Small 2	Complex 1	Easy 2	High 2	High 1

[1] The higher score like 2 means better than 1.

Considering the requirements of a dexterous hand, modularity is a key feature that makes usage and maintenance more convenient. Referring to other dexterous hands that have experienced issues during use, such as broken finger joints, tendon failure, and motor failure. These issues are mostly related to the finger part, which is the most fragile part of a dexterous hand. Designing the hand with a component module and designing the finger module for easy disassembly allows users to replace the whole finger module without complex operations during maintenance. The final design takes the whole finger as a component module to realize a user-friendly design.

In designing the finger module, the choice of actuation method for the finger joint is a crucial factor. Referring to different solutions in other dexterous hands [Ma et al., 2013a], various solutions exist to meet the requirements of full actuation, including tendon-driven, linkage, and gearbox designs, which are shown in Figure 3.3.

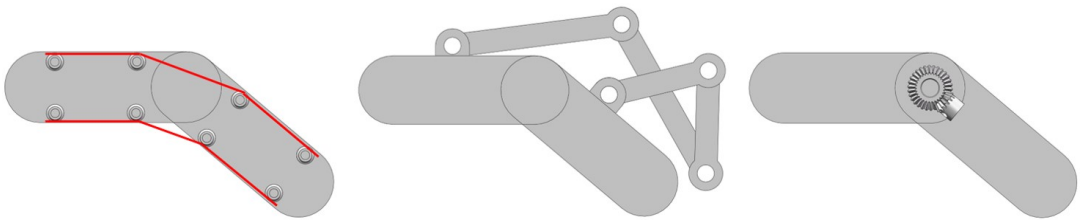


Figure 3.3: Three types of actuation method

Tendon-driven designs are popular as they mimic the mechanism of the human hand. They have the advantage of being lighter at the end of the hand and can provide faster motion. However, they have a more complex tendon structure, requiring more maintenance, especially due to material inelastic deformation and wear problems. Given the goal of creating a design that is easy to use and maintain, it was not

chosen for this solution.

Linkage solutions have become popular in recent years, particularly in the field of brushless motors and legged robots. They offer the advantage of good explosive force and quick reaction. However, the linkage structure occupies a large space and needs enough space for movement. This factor affects the flexibility and adaptability of the finger design. A requirement with one DOF is more suitable for such a solution.

The gearbox design is the most traditional and widely used solution. It provides enough torque and strength to the joint with a certain gearbox. This approach has been chosen to ensure the joint is both slim and powerful. Even though it is a traditional solution, it still poses challenges, such as being lightweight, providing sufficient power, and being easy to use.

A comparison of these three solutions can be found in Table 3.2. The gearbox solution with a simple design and limited cost was selected for the final design.

Table 3.2: Comparison of different actuation methods

Solution	Joint volume	Total volume	Structure	Maintenance	Cost
Tendon-driven	Small 3	Medium 2	Complex 1	Complex 1	Medium 2
Linkage	Big 1	Big 1	Medium 2	Medium 2	High 1
Gearbox	Medium 2	Small 3	Simple 3	Simple 3	Low 3

[1] The higher score like 2 means better than 1.

3.1.1.2 Hand Layout

The hand layout is mainly concerned with the allocation of DOF and the positional relationship between the fingers and palm. This is important for a dexterous hand because the layout influences the hand's gestures and grasp range, which are directly related to its dexterity.

The number of fingers is the first factor that influences the layout. From the perspective of force analysis, three fingers are the basic requirements for an end-effector that can manipulate an object in three-dimensional space without the external environment and force, and it is the most economical design. Therefore, this thesis mainly focuses on the design of a three-finger dexterous hand with fully-actuated DOF.

The layout of the finger location is important for the relationship among the three fingers. A human-like layout is widely used in bionic hands to mimic the function of the human hand and make robotic actions more human-like. However, with this layout, the joint motion of the thumb becomes complex when three fingers are needed to form a force-closure status. To simplify the design of the thumb base, the layout can be simplified by moving the thumb location to the middle of the other two fingers. The layout change process can be seen in Figure 3.4. In these layouts, all three fingers

have the same function, and the middle finger can be regarded as the thumb. The complex motion of the thumb base can be replaced by the motion of the robot arm. Then, the thumb base can be set as fixed with the palm, and the two DOF in the palm are with the other two fingers.

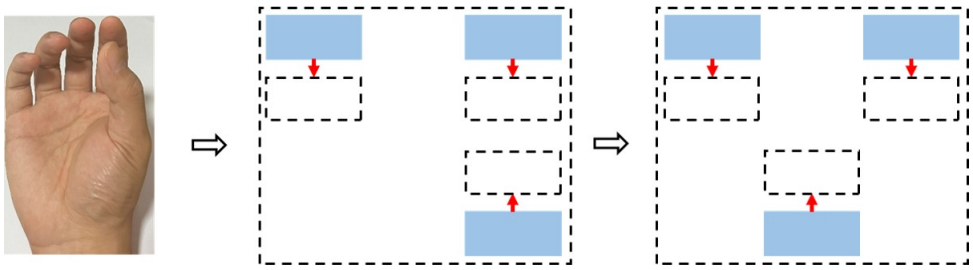


Figure 3.4: The human hand layout and the layout designs

With the defined layout and DOF distribution, the remaining issue is the DOF type for the two finger bases. Figure 3.5 shows three potential layouts for the dexterous hand. Each layout has different advantages, which can be compared in Table 3.3.

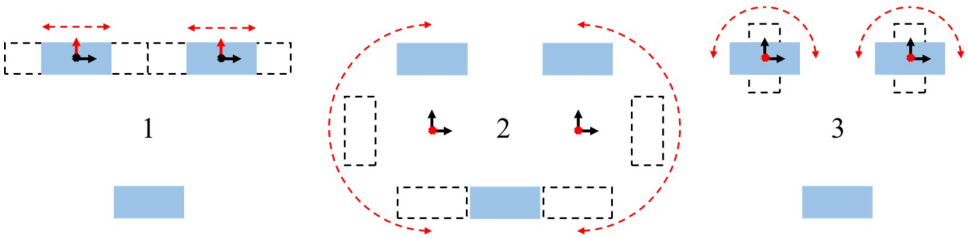


Figure 3.5: Three layouts of the three-finger hand

Table 3.3: Comparison of three types of layout

Layout	Volume	Gesture number	Tiny object	Grasp range	Strength
Layout 1	Small 3	Low 1	Medium 2	Small 1	High 3
Layout 2	Big 1	High 3	Hard 1	Big 3	Low 1
Layout 3	Medium 2	Medium 2	Easy 3	Medium 2	Medium 2

[1] The higher score like 2 means better than 1.

Layout 1 mimics the DOF in the human finger MCP. It can only perform parallel grasp with three fingers. Tiny objects can be grasped by decreasing the gap between two fingers. This layout has been used in the Robotiq three-finger hand [Robotiq, 2008]. Layout 2 has the largest layout and can grasp larger items. The distance under the two-finger model is bigger and can accommodate three fingers on one side. The main drawback is that the long arm of the two fingers may be fragile and there may be issues with backlash and synchronization. This layout has been used in

the BarrettHand [Townsend, 2000]. Layout 3 provides a small layout, and the motion at the base of the two finger bases can provide angle change. It can also change to a two-finger version. A similar layout has been used in the Schunk three-finger hand [Schunk, 2008].

After comparing different layouts, two dexterous hands with layout 2 and layout 3 have been developed. Layout 2 has been used for the one named Eagle Shoal [Wang et al., 2019a], which has more diverse gestures and is aimed to be used in a domestic environment. Layout 3 has been chosen as the ideal candidate for the three-finger DoraHand [Wang et al., 2022], which offers a suitable range of gestures and a more stable structure, and is aimed to be used in an industrial environment.

3.1.2 Sensing Capability

The sensing capability of a dexterous hand is crucial, as it determines the type and amount of data that can be collected. This directly influences the function and performance of the hand. The sensing capabilities of a dexterous hand can be divided into two types: 3.1.2.1 internal sensing and 3.1.2.2 external sensing.

3.1.2.1 Internal Sensing

Internal sensing refers to sensors that provide internal data of the dexterous hand and are not directly affected by external factors. Common examples include angle, current, and acceleration sensors, which are essential for controlling the hand's motion and monitoring its status. For this dexterous hand, joint angle, current, and temperature sensors were chosen as internal sensors due to their maturity and numerous references.

The joint angle sensor measures the joint angle, and various types are available in the industry, such as capacitance, resistance, optical, and hall effect sensors. To ensure a longer lifetime, easier assembly, and simpler maintenance, a non-contact sensor is preferred. In this design, the hall effect sensor is a good choice. Its basic mechanism can be seen in Figure 3.6.

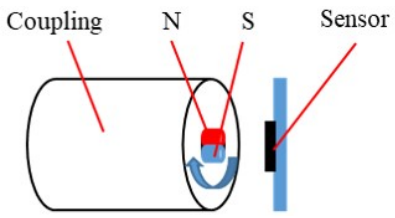


Figure 3.6: Mechanism of hall effect sensor

The joint angle provided by the hall sensor is used for position and velocity control.

To enhance control performance, current control is added to the inner control loop. The current sensor is mounted on the control board. A temperature sensor is used to protect the motor from potential over-temperature risks inside the joint.

3.1.2.2 External Sensing

External sensing refers to sensors that provide external data for the dexterous hand, including visual, tactile, force, torque, and other sensors. Visual sensors such as the RGB and RGB and Depth (RGBD) sensors can be mounted either inside or outside the hand to provide visual data during the grasping process. Considering the potential of visual occlusion and the performance of different visual sensors, mounting the sensor outside the hand is often a better option. The mechanical interface for the visual sensor is designed on the base of the hand.

Tactile feedback is essential for informing the robot about its interactions with target objects and is crucial for performing robotic manipulation. Tactile sensing in robots is defined as the process of detecting and measuring a given property of a contact event in a predetermined area and subsequent preprocessing of the signals at the sensor level itself – before sending them to higher levels for perceptual interpretation [Dahiya et al., 2009, Dahiya and Valle, 2008]. According to this definition, tactile sensing data can be obtained using various sensors that can detect contact events. For a robot component, tactile feedback can be generated with various data such as force, pressure, visual, and others.

There are numerous sensors available that can provide tactile feedback, including those based on mechanisms such as the piezoelectric effect, air pressure, electromagnetic fields, visual, capacitance change, and piezoresistive effect. These different sensors need to be analyzed to determine which is most suitable for this application.

Sensors that utilize the piezoelectric effect are commonly used in weight scales and other high-precision measurement instruments. These sensors detect micro deformations within their structure, and use the relationship between force and deformation to calculate the force value and direction. One advantage of this type of sensor is that its structure can be modified according to different designs, and can sense six-dimensional force and torque. However, one drawback is that the structure is usually large to provide enough space to mount the sensor. This type of sensor can sense the force applied to the structure but cannot sense where it is applied. This makes it more suitable for use at the wrist between the hand and the robot arm. Figure 3.7 shows a popular sensor [ATI, 1989].

One type of sensor that employs an air pressure mechanism is a viable option. These sensors are constructed with a hermetic space that contains an air pressure sensor and sufficient empty space. When touched, the outer surface of the hermetic

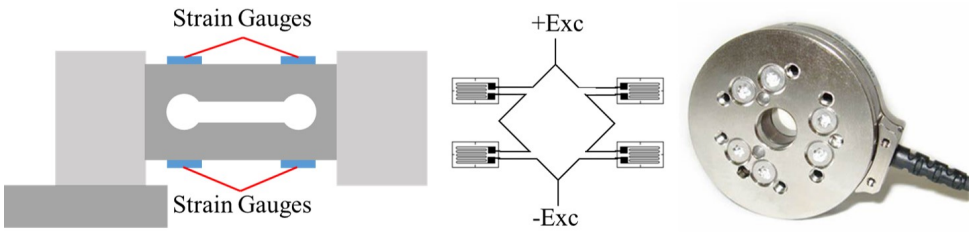


Figure 3.7: Strain gauges, measurement circuit, and six-axis F/T sensor

space deforms, causing a change in pressure within the space. The sensor can then measure the value of this pressure change. With calibration, the sensor can provide the force value. This type of sensor is sensitive and can take advantage of mature air pressure sensor technology. However, there are two main considerations with this type of sensor. First, the material used to construct the sensor should be soft enough to cause a sufficient pressure change, but the force value may be influenced by the deformation of the soft material. Secondly, the soft material on the sensor limits its thickness and it cannot be too small. Figure 3.8 (a) shows a typical air pressure sensor TakkTile [Tenzer et al., 2014].

Sensors with an electromagnetic mechanism utilize magnetism to detect the position of the top layer and convert it into force and deformation measurements. Like air pressure sensors, this type of sensor requires a soft material, but the limited material and dimension can restrict the design of the hand. Figure 3.8 (b) shows a typical electromagnetic sensor ReSkin [Bhirangi et al., 2021].

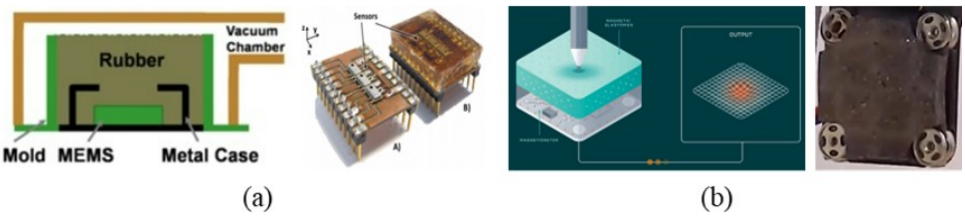


Figure 3.8: TakkTile and ReSkin sensors

Visual sensors are currently a popular type of tactile sensor. These sensors provide a visual image of the touch surface, which can facilitate research on the influence of texture. This type of sensor has evolved from the original GelSight [Yuan et al., 2017] to many different designs with similar mechanisms, including GelSlim [Donlon et al., 2018], Digit [Lambeta et al., 2020], and others. These sensors can provide enough texture information and the touch position can be clearly displayed. The force can be calculated through the calibration of visual and deformation images. The visual output of these sensors also offers a more convenient way to combine visual and tactile information. However, one main concern with this type of

sensor is its large size. There must be a distance between the touch surface and the visual sensor, even when using a reflector. This volume issue limits the hand and makes it unsuitable for most cases. Some typical sensors are shown in Figure 3.9.

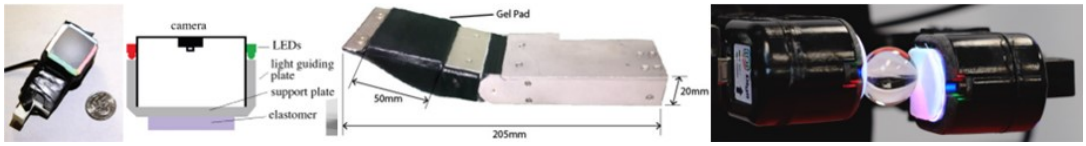


Figure 3.9: GelSight, GelSlim, and Digit sensors

Sensors based on capacitance change and the piezoresistive effect are traditional sensors with simple mechanisms, as shown in Figure 3.10. Capacitive sensors detect changes in capacitance between two parts and the force can be calculated through calibration. These sensors can be manufactured with a thin dimension. Although the capacitance change of one sensor only provides one value, a high-density array can be manufactured for planar force sensing. Piezoresistive sensors are based on the relationship between resistance and force. The force causes the micro deformation of the sensor surface, resulting in a change in resistance. The linearity of the piezoresistive sensor output is good and its properties are stable. Sensor arrays for planar sensing can also be produced. The most famous tactile sensor with this mechanism is the BioTac shown in Figure 3.11 [SynTouch, 2007, Su et al., 2012].

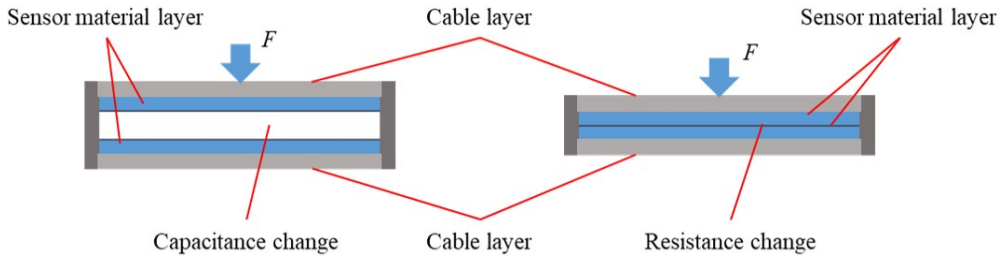


Figure 3.10: Mechanism of capacitive and piezoresistive sensors

After comparing these options, capacitive and piezoresistive tactile sensors with advantages in dimension, design compatibility, and cost, were determined to be the better choice for this design. Considering actual manufacturing and production levels, the piezoresistive sensor was selected for use in the tactile module.

3.2 Key Components Design

Building on the concepts discussed in the previous section, the design process requires further exploration and comparison of various solutions and components. This section presents the design of key components and consists of three subsections: 3.2.1

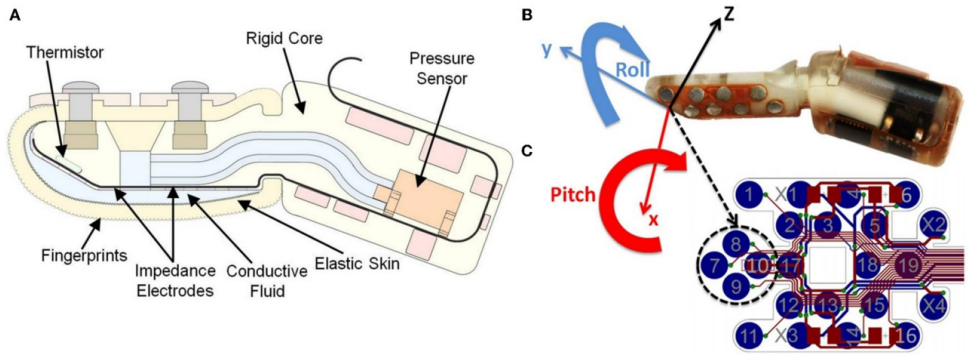


Figure 3.11: Sensor structure inside BioTac

modular finger, 3.2.2 sensor module, and 3.2.3 tip function and shape.

3.2.1 Modular Finger

To guarantee finger performance and achieve extensibility, it is important to design a finger with a small dimension, sufficient motion range, and sufficient joint power. Some key considerations include joint range, joint layout, finger structure, and other structures. These factors contribute to the design of a finger that is both functional and scalable.

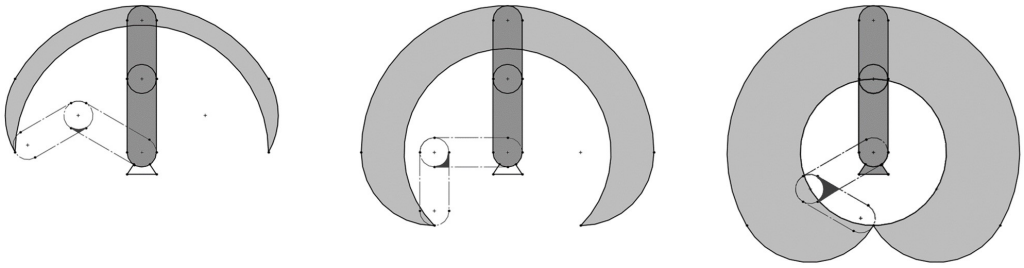


Figure 3.12: Joint range and fingertip motion range

Joint range As discussed in the subsection 3.1.1, the finger module has two DOF and the motion range of each finger is important. As shown in Figure 3.12, a larger motion range improves the manipulation capability of the finger, but also increases the dimension of the intersection area, reducing the volume of the base knuckle and space for the motor. To determine the motion range of the finger joint, it is necessary to measure the motion range of a human finger, especially during the grasping task. Refer to the joint range of human finger joints in Table 3.4 [Bain et al., 2015], the maximum angle is the active flexion angle of PIP at 101°, and the minimum negative value of MCP at -19°. The joint range during the grasp task is from -5° to 93°. Based on these values, and considering that the negative motion range has no significant

influence on finger design, the motion range of the finger joint can be set at ± 90°. This range can already cover all the range required for grasping tasks, and the negative motion range can help realize special gestures and leave enough space to avoid obstacles.

Table 3.4: Human finger joint motion range under different situations

Joint	Active/°		90% Activities Functional/°		Pre-grasp & Grasp Functional/°	
	Extension	Flexion	Minimum	Maximum	Minimum	Maximum
MCP all	-19	90	19	71	-5	77
PIP all	-7	101	23	87	16	93
DIP all	-6	84	10	64	1	72

Joint layout The position of the motor is crucial in the joint layout, and there are two methods of mounting the motor. These two designs are used in the Eagle Shoal and DoraHand versions respectively. The main difference between these designs is the number of parts in the finger structure: two or three. The two-part design has a smaller total volume and lower integration cost, allowing for a smaller total volume of the hand. The three-part design has no motor in the front part, allowing its shape to be designed to meet specific requirements and be replaced with other components. The three-part version is more practical during usage and research. The comparison figure and table can be seen in Table 3.5 and Figure 3.13.

Table 3.5: The main difference between the two designs

Layout	Volume	Output force	Fingertip dimension	Extension
Eagle Shoal	Small 2	Small 1	Big 1	Hard 1
DoraHand	Big 1	Big 2	Small 2	Easy 2

[1] The higher score like 2 means better than 1.

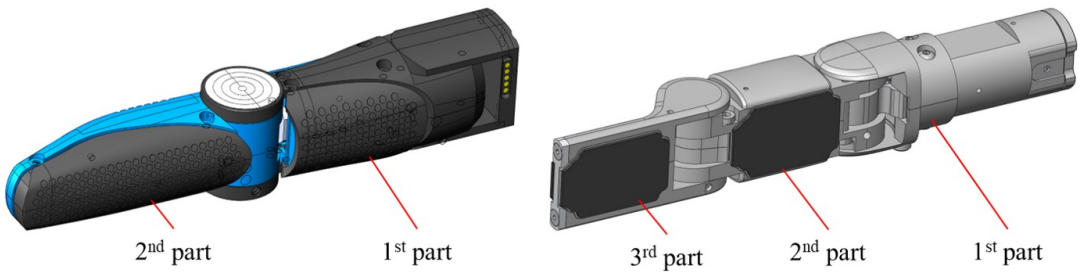


Figure 3.13: Comparison of two-part and three-part fingers

Finger structure When designing the finger joint, careful consideration must be given to the selection of the motor and reducer. The length of each knuckle is crucial

and can be determined by referring to human fingers lengths. In this design, the DIP and PIP are merged into one part, and the length of these two knuckles in the finger should be similar, and set with a value around 50 mm. For the joint motion parameters, referencing other dexterous hands, the fingertip force is set at around 20 N, and the finger joint motion speed is set at around 60°/s. To account for the different torque requirements of each joint, the torque added on joint 0 and joint 1 can be calculated as shown in Equation 3.1 and Figure 3.14.

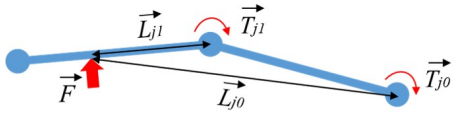


Figure 3.14: Torque on different joints

$$\begin{cases} \vec{T_{j0}} = \vec{L_{j0}} \times \vec{F} \\ \vec{T_{j1}} = \vec{L_{j1}} \times \vec{F} \end{cases} \tag{3.1}$$

In Equation 3.1, $\vec{F}$ represents the external force, $\vec{L_{j0}}$ and $\vec{L_{j1}}$ represent the distance from external force to joint 0 and joint 1, respectively, and $\vec{T_{j0}}$ and $\vec{T_{j1}}$ represent the torque on joint 0 and joint 1, respectively. This calculation shows that the maximum torque on joint 0 can be twice that of joint 1. To ensure the finger joint speed is not too slow, the motors in joint 0 and joint 1 should be different, with the output torque of joint 0 being more than twice that of joint 1.

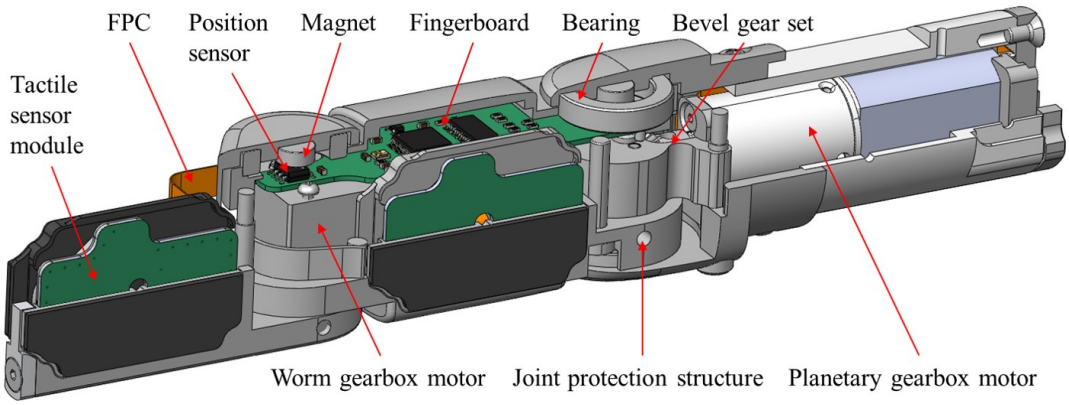


Figure 3.15: Structure of finger module

Due to the limited volume in the finger knuckle, a worm gearbox is selected for joint 1, as it can be smaller and output greater torque. A planetary gear reducer and a set of bevel gears are used in joint 0 to transmit the torque. With the selection of the gearbox and motor, the main part of the finger joint is decided. Each finger module

includes two motors and two force sensors, with two sensor modules mounted inside the two knuckles. The whole structure of the finger module can be seen in Figure3.15.

Other structures Another important factor to consider is the structure inside the joint. The gearbox with a large reduction and self-locking mechanism can help maintain the joint position and save power during grasping. However, the self-locking mechanism can also lead to accidental injury when external torque exceeds the maximum bearing torque of the joint. To address this, a protection structure is necessary for each joint. This structure uses a spring plunger to limit external force. When the torque exceeds the joint's maximum bearing torque, the plunger releases and the joint enters free-moving mode, separating from the actuation module. The force analysis scheme can be found in Figure 3.16. The force is calculated using the Equation 3.2, where T_{max} represents the joint's maximum bearing torque, F_T represents the torque force on the coupling, the r_c represents the radius of the coupling, the F_p is the plunger's spring force, the θ represents half the angle of the mating groove on the coupling. This design significantly extends the joint's lifespan.

$$T_{max} = F_T r_c = F_p r_c / sin(\theta) \tag{3.2}$$

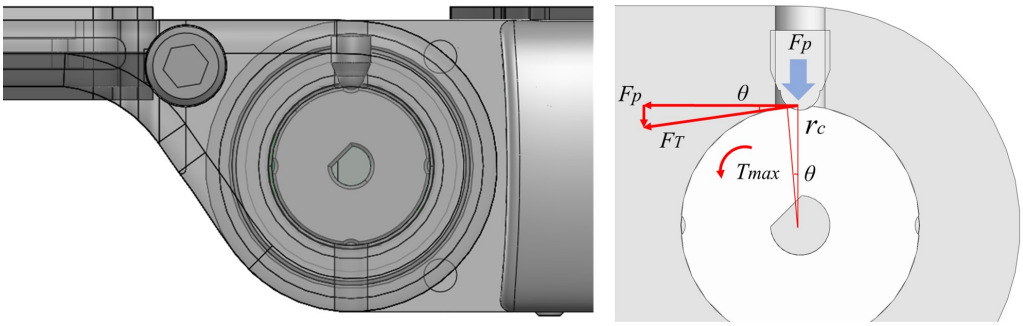


Figure 3.16: Scheme of force between joint and coupling

Furthermore, to improve sensor data acquisition and processing and enhance control, joint angle and finger boards are located inside the finger module with minimal space requirements as shown in Figure3.15. This approach enables the finger module to function as an independent unit, contributing to the overall modularity and flexibility of the design.

3.2.2 Tactile Module

As mentioned in subsection 3.1.2, the tactile sensor plays a crucial role in enabling the robot to perform manipulation tasks. Compared with integrating commercially available tactile sensors, incorporating a tactile module within each component is

more stable and guarantees contact with objects. This subsection introduces the tactile module in two steps: 3.2.2.1 sensor layout, and 3.2.2.2 tactile module design.

3.2.2.1 Sensor Layout

Due to space limitations and production difficulties, film sensors utilizing the piezoresistive effect mechanism are a suitable option for integration within the hand design. Instead of building a big sensor with a large area, using a sensor array to form a sensing plane and obtain a large sensing area is a common method. Existing literature shows that sensor arrays are widely used for data feedback in robot hand designs [Yousef et al., 2011, D. et al., 2009]. The common sensor array is typically sparse and covered with a top surface, as depicted in Figure 3.17. To achieve a better sensor design, two key points that need to be considered.

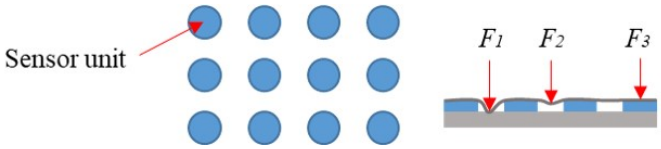


Figure 3.17: Tactile sensor array

One key consideration is the top surface for the sensor array. Without a top surface, sensors in the sensor array are independent and gaps exist between adjacent sensors, regardless of the number of sensors in the array. When the robot hand grasps a tiny object, only sensors in direct physical contact provide data output, leading to data loss due to discontinuity. To address this, a surface is needed on top of these sensors to maintain contact and distribute force. Surfaces with both soft and rigid materials are available. A soft surface guarantees contact between the surface and sensor and supports multi-point contact, but deformation may cause sensing force to change. A rigid surface ensures accurate force distribution but can only sense the position of combined forces.

Another consideration is the relationship between sensor area and precision. Reducing the area of one sensor also reduces its contact surface, leading to unexpected contact. Unstable contact with a small area leads to bigger errors for the film sensor. Therefore, the design of the sensor module should prefer a solution that can sense continuous force and has a stable sensing area. A rigid surface and large area sensor are applied in the design.

After conducting extensive experiments, the film force sensor I-MOTION IMS-C10 shown in Figure 3.18 was chosen. It has a small thickness of 0.22 mm and a range of 10 N with a resolution of 10 mN. The detailed design of the sensor module is introduced next.

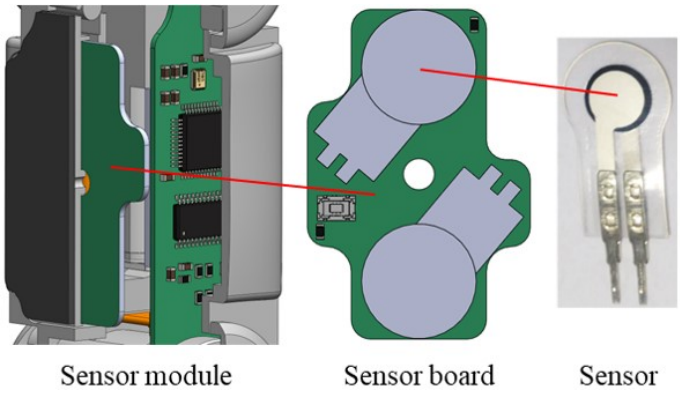


Figure 3.18: The sensor module, sensor board, and film sensor

3.2.2.2 Tactile Module Design

The design of the sensor module structure is crucial for achieving continuous force detection. Since three points can determine a plane, the force distribution on the surface of the contact points can be determined using fewer sensors. As shown in Figure 3.19, two sensors are used in each finger knuckle module, which is limited by the width of the finger, and four sensors are used in the palm area.

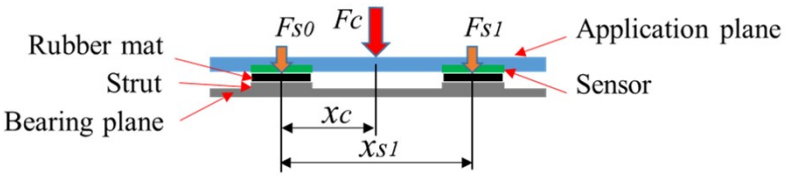


Figure 3.19: Scheme of the sensor module

A force application plane with struts structure and rubber mats shaped to fit the effective perception area ensures stable contact with the sensor. The application plane can transmit force to the sensors through the struts, regardless of the contact point with the object. The combined force and combined contact point position can be calculated using Equation 3.3. The force F_c is the combined external force, the F_{s0} and F_{s1} are the component forces on the struts, and the m is the number of component forces. The x_{s0} and x_{s1} are the distances from strut $i = 0, 1$ to the origin point, and the n is the number of struts. The x_c is the position of the combined force in that direction, and the v_{xc} is the velocity of the force moving in the x direction.

$$\begin{cases} F_c = \sum_{i=0}^{m} F_{si} \\ x_c = \sum_{i=0}^{n} F_{si} x_{si} / F_c \\ v_{xc} = x_c' \end{cases} \tag{3.3}$$

To sense force in a plane and enhance its capabilities, the palm module has been designed with four sensors, as illustrated in Figure 3.20. The coordinates in the palm module are set along the y and z axes, corresponding to the x axis in the finger module. The combined force F_c is calculated as the sum of F_{si}, and the position y_c and z_c are determined using the positions y_{si} and z_{si} of the struts. The velocities v_{yc} and v_{zc} are also calculated using these position values. The F_{s0}, F_{s1}, F_{s2} and F_{s3} are component forces on the four sensors. The combined force position and velocity can be calculated using Equation 3.4.

$$\begin{cases} F_c = \sum_{i=0}^{m} F_{si} \\ y_c = (F_{s0}y_{s0} - F_{s2}y_{s2})/(F_{s0} + F_{s2}) \\ z_c = (F_{s3}z_{s3} - F_{s1}z_{s1})/(F_{s1} + F_{s3}) \\ v_{yc} = y_c' \\ v_{zc} = z_c' \end{cases} \tag{3.4}$$

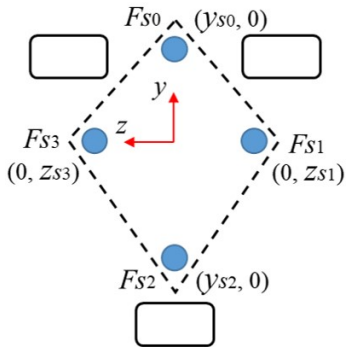


Figure 3.20: Structure of palm layout

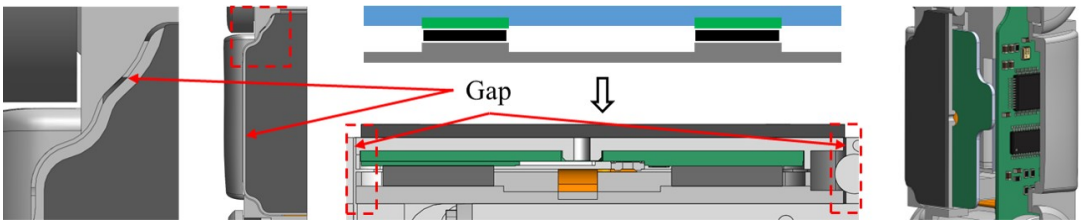


Figure 3.21: The detailed structure of the sensor module

All of these calculations are based on the assumption that the contact between the sensor and strut is optimal and their relative positions are fixed. However, if external forces are not only vertical, the resulting change in position can also affect sensor accuracy. To address this issue, rubber is used to increase the contact area and is attached to the sensing area to maintain a fixed relationship. The rubber is

pasted on the sensor to ensure a fixed position between the rubber and the sensor. For the relationship between the rubber and force application plan, unlike solutions that involve adding preload or fluid, keeping a gap between the rubber and force application plane can reduce maintenance difficulty. This gap is controlled with a minimal margin to avoid significant errors. These details are crucial in the module design and can be seen in Figure 3.21.

With this design, the sensor unit can continuously detect force and significantly aid in sensing slip motion. Additionally, a larger sensor array can be replaced by fewer sensors, reducing the total costs.

3.2.3 Tip Function and Shape

The tip of the finger module is important as it can perform a variety of functions. A well-designed tip with specific shape and extension capabilities is critical for accomplishing various tasks. This subsection introduces the design of the tip function and shape.

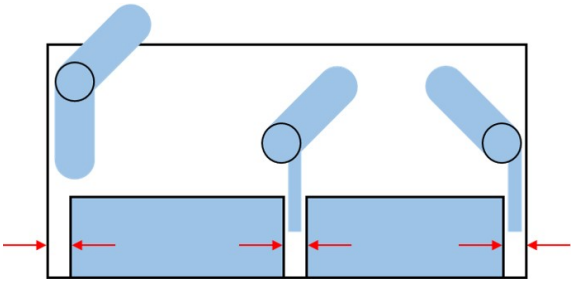


Figure 3.22: Grasp pose in bin-picking

Compared to common scenarios such as grasping on a table or shelf, grasping items inside a bin is more challenging for the end-effector. Considering the bin-picking scene can help improve its capability. A hand used for bin-picking should have a thin tip and be able to perform various gestures, as shown in Figure 3.22. These gestures facilitate tip manipulation tasks and allow fingers to access narrow spaces, especially in bin-picking. The fingertip is designed to meet this requirement by being as thin as possible. The hard surface in the tip area, combined with the rubber, is inspired by the human finger, which consists of the nail and finger pad. This design can adapt to complex environments and make grasping easier and more stable.

As the finger base joint range in the palm is $\pm$ 90°, the two fingers can grasp items with both sides of the fingertip. It is important to have the sensing capability on both sides of the tip area. To achieve two-sided sensing of the sensor module, a special structure is designed in the tip that keeps a gap between the module and separates

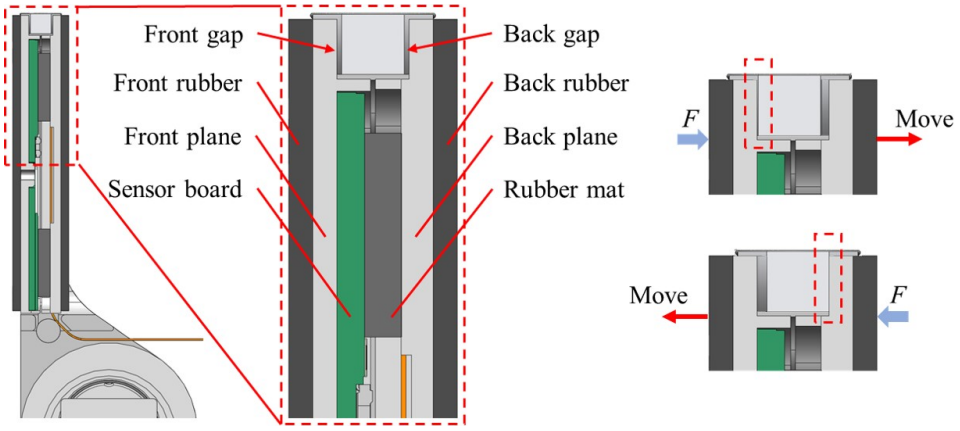


Figure 3.23: Structure of fingertip and force diagram

the sensing part into two sections. When force is applied from the front or back, it is transmitted through the sensor strut and added to the support structure. This sensor module helps the tip to sense in both directions. The tip structure and force diagram can be seen in Figure 3.23.

For dexterous hand usage, researchers often have diverse hardware requirements, particularly when new sensors need to be evaluated, markers need to be added, or structures need to be replaced [Kawasaki et al., 2002]. The tip is designed with an easily disassembled structure that can be extended or replaced with other components. The tip structure can be seen in Figure 3.24.

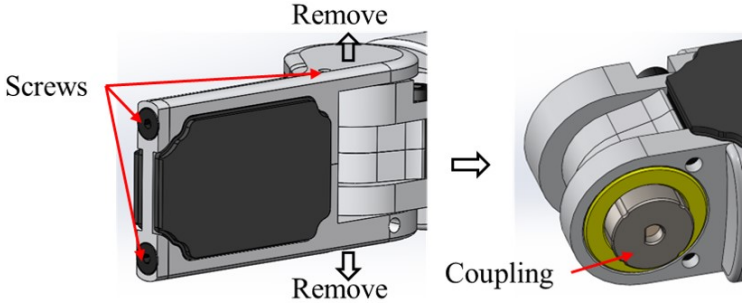


Figure 3.24: Structure of fingertip

3.3 Whole System Design

With the design concept and key components in place, it is necessary to complete the whole system design from a full-view perspective. The application industry and scenarios play a decisive role in determining the payload capacity of the dexterous hand. Collaborative robots with payloads ranging from 5-10 kg have been widely adopted to ensure safety and cooperation with workers. Considering the estimated weight of

the dexterous hand to be around 1-2 kg, a payload capacity of 6 kg is suitable and can cover most grasping and manipulation requirements. Based on this application requirement, the design of the dexterous hand can be divided into three subsections: 3.3.1 mechanical design, 3.3.2 electronic layout, 3.3.3 embedded and application software.

3.3.1 Mechanical Design

The mechanical design covers all structural designs, including the components presented in the previous subsections. This subsection has two additional sections to discuss: 3.3.1.1 modular hardware interface and 3.3.1.2 dimension and layout.

3.3.1.1 Modular Hardware Interface

With the designed finger module, the modular hardware interface serves to connect the finger module to the palm. There are two basic requirements for the modular hardware interface: reliable connection and easy assembly.

Inspired by connectors used in consumer electronics, the design utilizes Pogo Pins, which are spring-loaded pins, for stable electrical connection. The design uses spring plungers and magnetic connectors for reliable mechanical connection, where the spring plunger provides force and location functionality, and the magnetic connection enhances the connection force. The force diagram between the finger and palm can be checked in Figure 3.25.

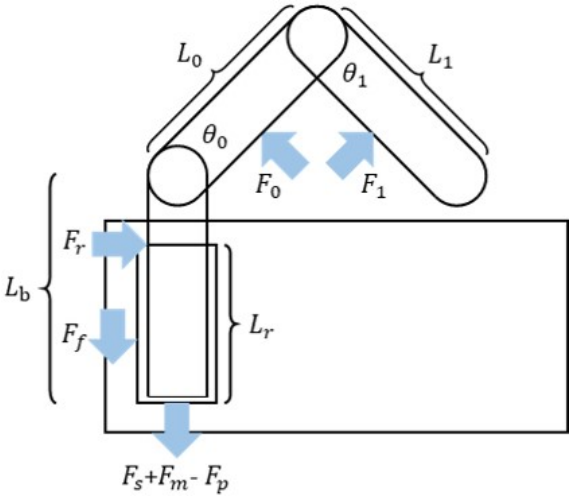


Figure 3.25: Scheme of the force on the finger

With the force diagram, the force calculation process is presented in Equation 3.5, which includes forces acting on the finger. These forces include the combined force F_c, the spring plunger force F_s, the magnetic force F_m, the Pogo Pins force

F_p, the friction force F_f, and external forces F_0 and F_1 applied to fingers 0 and 1, respectively. Additionally, θ_0 and θ_1 are the angles of fingers 0 and 1, respectively, and μ is the friction coefficient between the finger base and palm. The reaction force F_r balances all forces in the horizontal direction, and L_r is the distance from the F_r point to the connection point. The other parameters include L_0 and L_1, which are the lengths of each finger knuckle, and L_b, which is the distance between joint 0 and the connection point. As indicated by the equation, horizontal component forces lead to a larger reaction force F_r and friction force F_f. Components are chosen based on a worst-case scenario in which an external force is exerted in a vertical direction.

$$\begin{cases} F_c = F_s + F_m - F_p + F_f \geq F_0 sin(\theta_0) + F_1 sin(\theta_0 + \theta_1) \\ F_f = \mu F_r \\ F_r = (F_0 L_0/2 + F_1 L_1/2 + L_0 F_1 cos(\theta_1) + L_b(F_0 cos\theta_0 + F_1 cos(\theta_0 + \theta_1)))/L_r \end{cases} \tag{3.5}$$

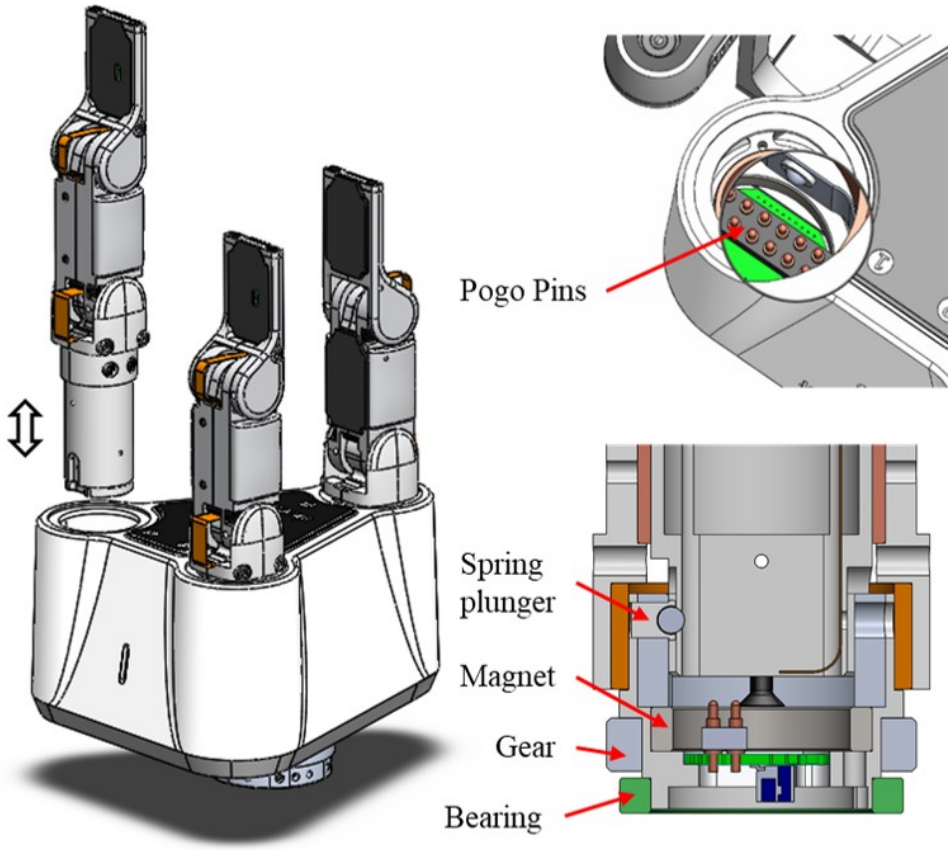


Figure 3.26: Structure of finger base

Based on this force analysis, suitable parameters for the spring plunger and mag-

netic connector can be calculated. With proper selection, the structure can withstand vertical plug forces of up to 3 kg, which is sufficient for a hand with a 6 kg payload. The structure is used for three finger bases, with two finger bases connected to two rotation bases inside the palm with a limited range of rotation, and the other one directly connected to the palm base. Each rotation base is driven by a motor through spur gears. The design is shown in Figure 3.26.

3.3.1.2 Dimension and Layout

The mechanical part must meet the previously mentioned requirements, and the layout should consider the dimensions of the target objects, such as those in the YCB objects set [Calli et al., 2015]. The common dimensions of daily life objects are usually limited to 100 mm, with the maximum dimension generally restricted to 150 mm. Given a finger knuckle length around 50 mm, the finger distance on the palm can be determined. The finger location is shown as an isosceles triangle and the coordinate definition is shown in 3.27.

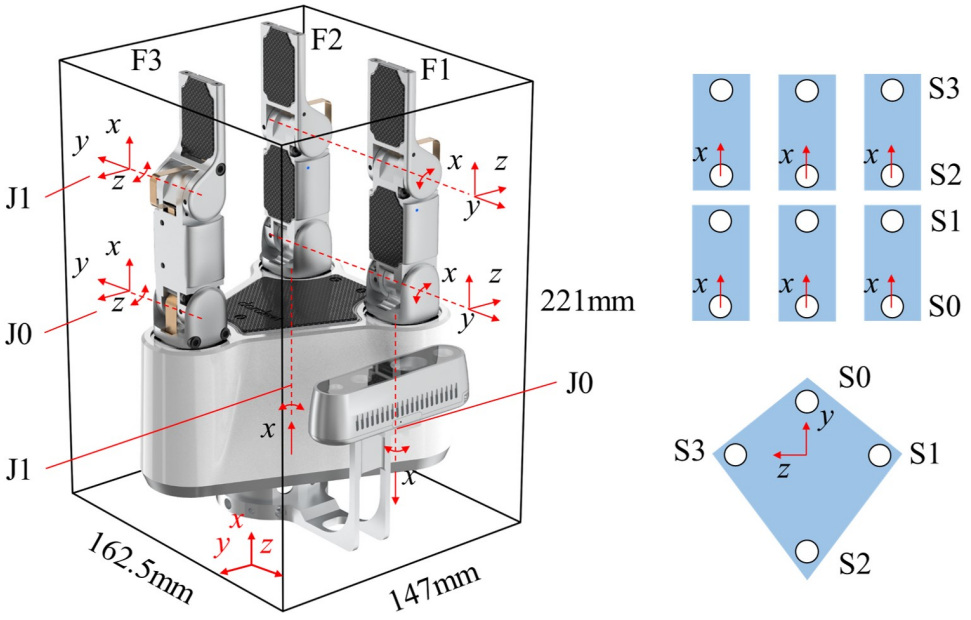


Figure 3.27: The dimension and coordinate definition of DoraHand

In addition to the layout, it is crucial to perform a strength analysis for each component during the design phase. Due to the considerations of self-weight, aluminum alloy is used in most parts of the hand, and plastic is used in some non-structural parts, such as the shell. Deformation of these components should be minimized and controlled within a small range to ensure proper functioning. Especially for the force application plane in the sensor module, the plane needs to maintain a tiny deforma-

tion to ensure accurate force results. After finalizing these details, the hand with various gestures is shown in Figure 3.28. Figure 3.28 (a) shows three different gestures with three fingers. Figure 3.28 (b) shows three different gestures with two fingers and with a positive grip. Figure 3.28 (c) shows three different gestures with two fingers and with a reverse grip. These nine gestures can be used as the basic gestures during grasping tasks.

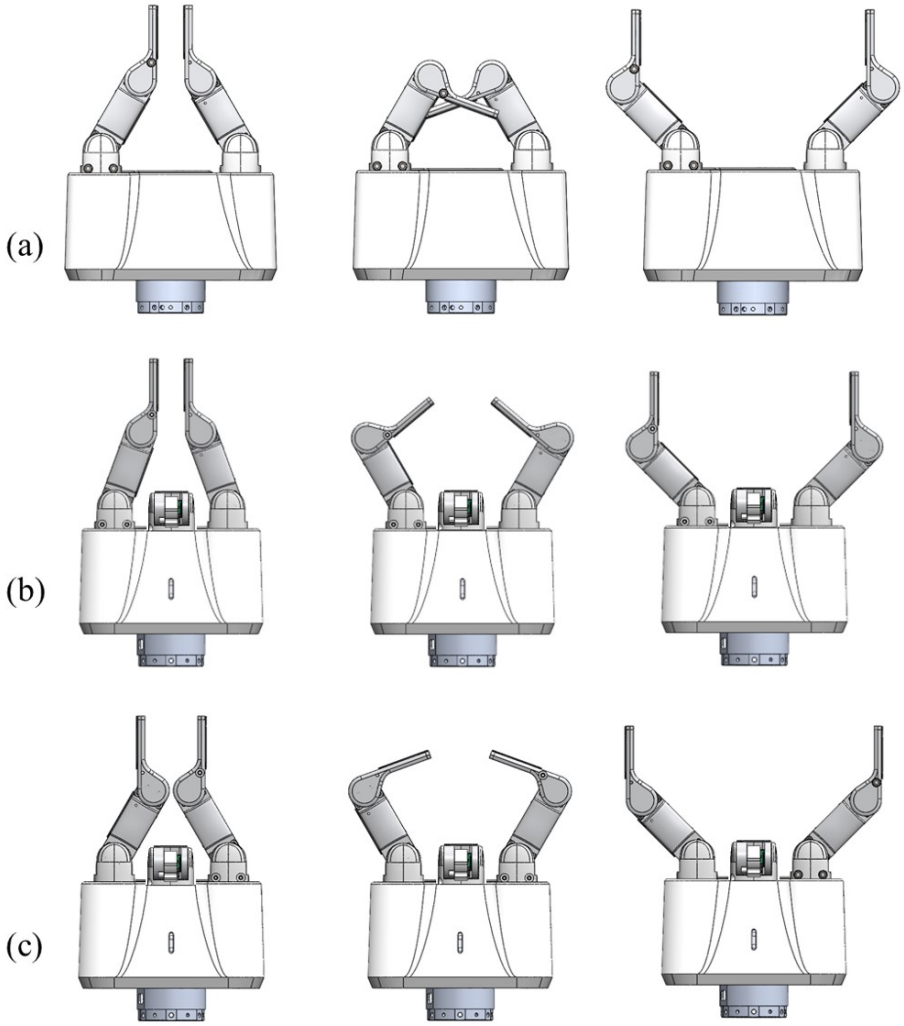


Figure 3.28: Different hand gestures with DoraHand

3.3.2 Electronic Layout

To make the module more user-friendly, each finger and the palm are equipped with a dedicated control board. Since the boards in each module have similar functions, system configuration is simplified. There is a finger board for each finger, a pico board for the palm, and a main board for the whole system. The finger board is responsible

for data collection, motor control, and communication. The pico board has a similar function as the finger board and can be expanded via onboard connectors. The main board is responsible for the communication and power supply of the whole system. The finger board, pico board, and main board are shown in Figure 3.30, and the overall structure can be seen in Figure 3.29.

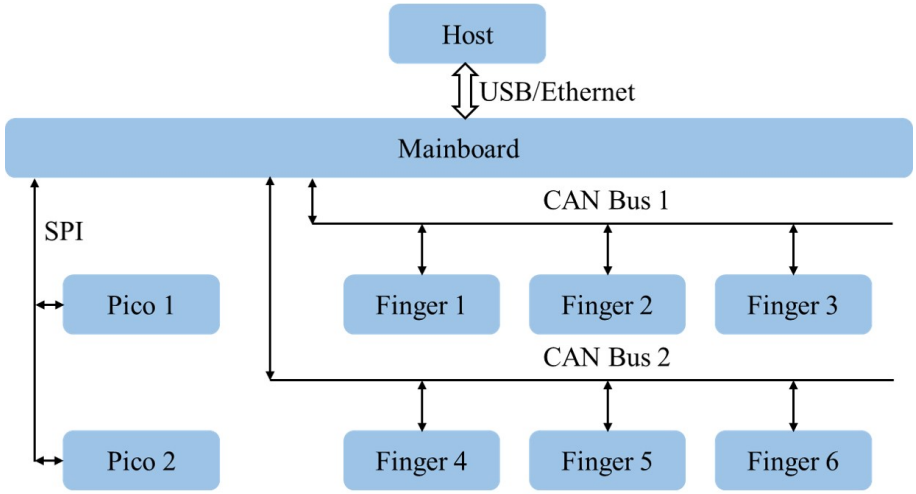


Figure 3.29: Structure of the electronic and embedded system

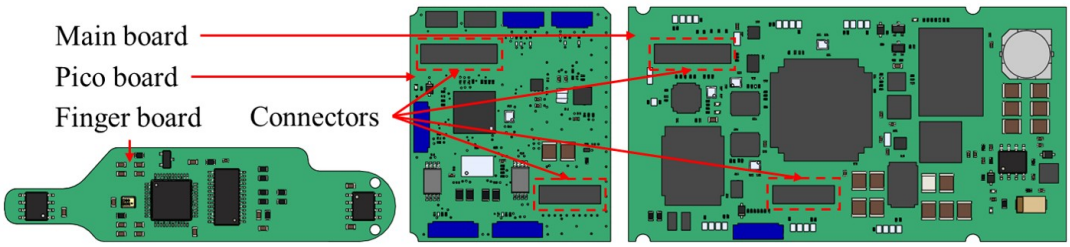


Figure 3.30: The finger board, pico board, and main board

The finger board is capable of collecting two joint position data, two current data, and four tactile sensor data with 10 bit and 1 kHz each. The Controller Area Network (CAN) bus protocol is employed for better expandability, with the MCP2515 and VP231 chips serving as the controller and transceiver respectively. Each board acts as an independent motion node within the hand system and is interconnected via Flexible Printed Circuit (FPC). In addition to the main board, a power module is embedded in the palm, receiving 24 V input voltage and outputting 5 V and 12 V.

To facilitate the connection of the finger module, a hot-swap function is developed for the hand. This feature ensures that fingers can be plugged in or out while the hand is powered and operating. In addition to mechanical design support, some changes have been made to the electronic design. When the finger module is in-

stalled, the power and communication of the finger module are activated, and when the finger module is not in place, the power and communication of the finger module are turned off. This is accomplished by detection circuitry in the main board. When a finger is connected, the detection voltage can be measured at the finger position, then starts and gradually increases the finger supply voltage, and then start communication. When not connected, the circuit remains open. Different detection voltages are enabled for each finger position and used for finger position recognition.

3.3.3 Embedded and Application Software

The Embedded and application software includes the work related to the embedded software and the corresponding application software on the user side. The communication framework is illustrated in Figure 3.30. The communication framework uses CAN communication between the finger boards and the main board, and Serial Peripheral Interface (SPI) between the pico boards and the main board. The hand has three fingers, with redundant ports for the addition of more fingers and actuators. Hardware trigger is utilized to ensure synchronization of each device's action time with multiple clients in the communication loop. Through the position, current and force data obtained by the board from the sensor feedback, the hand supports the loop control of position, velocity, current and force based on the integral-separated Proportional Integral Derivative (PID) method. Each module has been calibrated to meet requirements and can join the the CAN bus when plugged into place, allowing the main board to establish communication with any new fingers. The hand can communicate with other hosts via Universal Serial Bus (USB) or Ethernet, with a communication frequency of 60 Hz.

The application software includes a user interface with basic control functions for the dexterous hand, supporting data viewing and loop control with target commands. The interface visualizes sensor data values and shows the relation of force value and position in 2D drawing. A script mode supports script command execution, and a Control Performance Assessment (CPA) function assesses and adjusts joint parameters through a dedicated PID test module. The CPA function can help to adjust the joint parameter according to the running state and user requirements. Some main pages of the user interface can be seen in Figure 3.31.

In addition to the user interface, a library that supports communication via both USB and Ethernet is provided. Data can be subscribed through Robot Operating System (ROS) or ROS2 topics, and tasks can be executed through ROS or ROS2 services. The application software, library code, and Unified Robot Description Format (URDF) can be found on GitHub [Dorabot, 2021a].

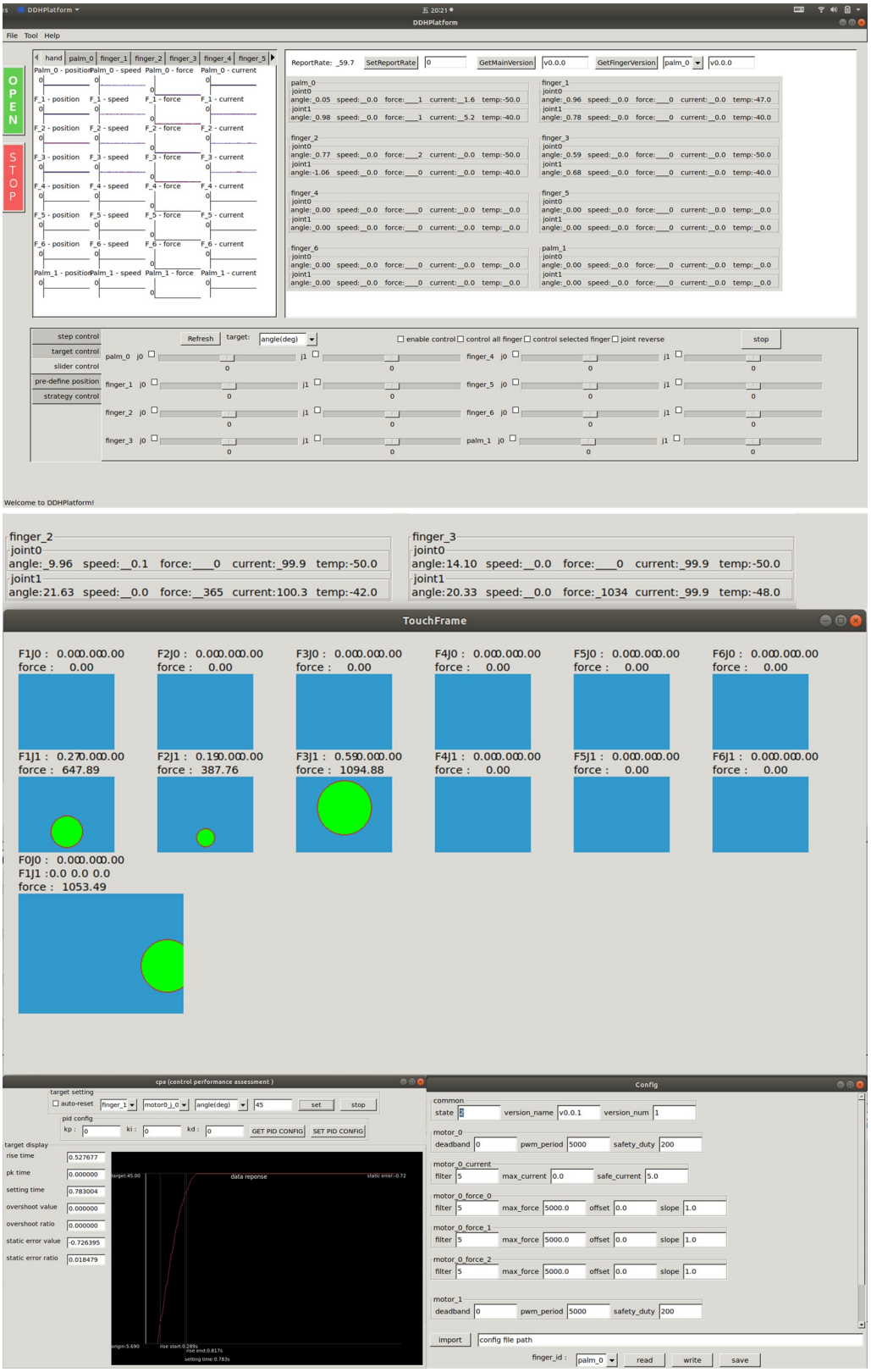


Figure 3.31: Some main pages of the user interface

3.4 Conclusion

The chapter provides an overview of the design of a fully-actuated modular three-finger dexterous hand product, discussing the concept and key design details. The tactile sensor module in the hand uses piezoresistive film sensors to provide excellent tactile sensing capabilities. The modular finger design has good scalability and maintainability, and a user-friendly interface enables hot-swapping of fingers. The hardware meets requirements for stable grasping and robotic manipulation, with sufficient DOF, tactile sensing, strong grasp force, easy-to-use, practicality, and affordability. The next chapter focuses on its verification and evaluation to explore potential applications.

Chapter 4

Experiment and Performance Evaluation

After designing the dexterous hands, Eagle Shoal and DoraHand, the hardware underwent several iterations during testing and usage. The latest version is the sixth iteration of DoraHand, which has addressed several drawbacks observed in different cases. This chapter aims to establish an evaluation procedure for dexterous hands and focuses on the experiments and performance evaluation of DoraHand. It is divided into four sections: Section 4.1 presents the basic parameters and capabilities of the hand. Section 4.2 showcases the capabilities of the sensor module. Section 4.3 introduces the work of extending to two-finger and five-finger versions of the dexterous hand. Section 4.4 provides a summary of this chapter.

4.1 Basic Performance

In the basic performance section, the fundamental parameters and grasping capabilities of the dexterous hand are demonstrated through experiments. This section includes two subsections: 4.1.1 experiment design and 4.1.2 performance.

4.1.1 Experiment Design

The basic parameters of a dexterous hand include its overall dimensions, weight, joint number and range, joint speed and precision, tactile sensor number and parameters, communication type and rate, grasping force, and capability. Most of these parameters are physical and can be easily measured using common measuring tools. The experiment design focuses on the measurement of grasping force and grasp capability.

As a basic parameter, grasping force does not have uniform criteria across differ-

ent hands due to variations in their layouts. The test uses two force parameters to demonstrate performance: encompassing grip payload and fingertip grip force. Encompassing grip payload is the maximum payload that the hand can grasp using a power grasp gesture. This value may be influenced by the shape, diameter, and friction coefficient of the target object. For this test, a cylinder with a diameter of 62 mm and enough strength to avoid visible deformation is selected, with a friction coefficient of around 0.6. The weight is changed by tying weights to the cylinder. Fingertip grip force is the maximum force output by the fingertip and can demonstrate the grasp force when using the tip. Grip force is measured using an electronic scale between the fingertips [Ma et al., 2013b]. The measurement states of two parameters are shown in Figure 4.1.

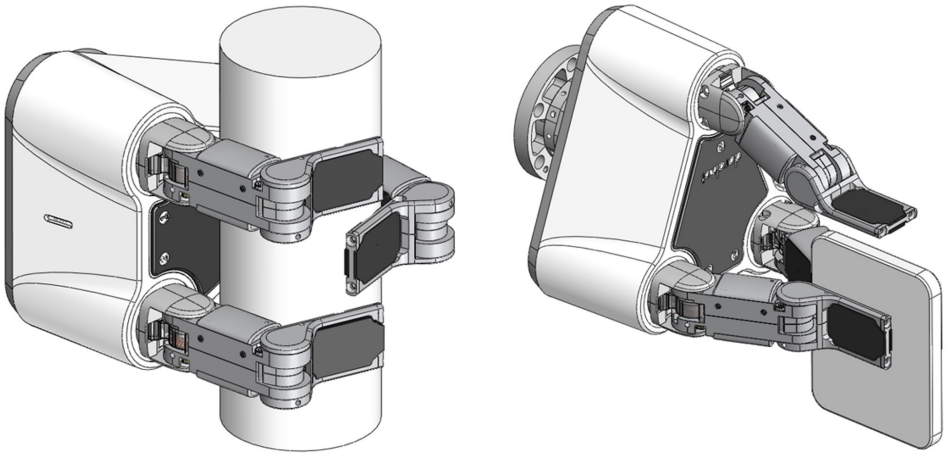


Figure 4.1: Two force parameter measurement methods

Grasp capability is also an important factor in evaluating basic performance. To test this capability, the hand is mounted on a robot arm, and the object is positioned for grasping. To standardize the evaluation criteria, the YCB object set was selected as the test set. This set includes 77 types of objects in 5 categories commonly found in daily life. The standard grasping procedure consists of four steps to grasp objects.

1. Reaching the pre-grasp pose.

2. Grasping the target object with position control and with no or tiny gap.

3. Using current control to ensure a stable and suitable force.

4. Controlling the robot arm with a translation motion for grasp stability verification.

Some special items not included in the object set were also added to the test, including cooked eggs with shells peeled off, circuit boards, and credit cards. These items can demonstrate the hand's ability to grasp soft and thin objects. For tools like scissors, the ability to manipulate them also needs to be tested.

4.1.2 Performance

The basic parameters of the hand can be calculated and obtained through control and designed experiments. These parameters are listed in Table 4.1. The joint sensor feedback provides a resolution of 0.01° and a precision of ± 0.1°. The encompassing grip payload of the hand is around 6 kg, and the fingertip grip force is around 25 N.

Table 4.1: Basic parameters of DoraHand

Dimension L-W-H	Weight	Finger weight	Joint number & range	Joint speed & precision
126 mm-143 mm-221 mm	1400 g	180 g	8 / ± 90°	70°/s / ± 0.1°
Sensor module number	Input power	Communication type & rate	Fingertip grip force	Encompassing grip payload
7	24 V / 2 A	Ethernet or USB / 60 Hz	25 N	6 kg

Compared with other dexterous hands, Table 4.2 shows a comparison of some main parameters [Ma et al., 2013a]. The DoraHand is the only modular hand with finger modules that support the hot-swap function. It has advantages in terms of DOF, weight, and payload. Its sensing ability can meet most requirements for robotic manipulation research and supports integration with other types of sensors and components.

Table 4.2: Main parameters comparison with different dexterous hands

Hand	Finger/actuator number	Grasp range/mm	Tactile sensor	Weight/g	Grip force/N
SDH Hand	3/7	239	Yes	1950	/
Barrett Hand	3/4	240	Yes	1200	15
Robotiq 3-Finger	3/2	155	/	2300	15-60
Allergo Hand	4/16	238	/	1500	/
BLT gripper	3/5	206	/	1200	10
Eagle Shoal	3/8	212	Yes	790	10
DoraHand	3/8	200	Yes	1250	25

The grasp capability test was conducted with the objects discussed in the experiment design. All objects in the test set can be grasped, and some objects can be grasped using several different gestures. The egg is stably grasped by the dexterous hand using the current control mode without damage. The scissors can be opened and closed by finger motion. These objects demonstrate that the DoraHand has good grasp and manipulation capabilities. Figure 4.2 shows fifteen typical sample results.

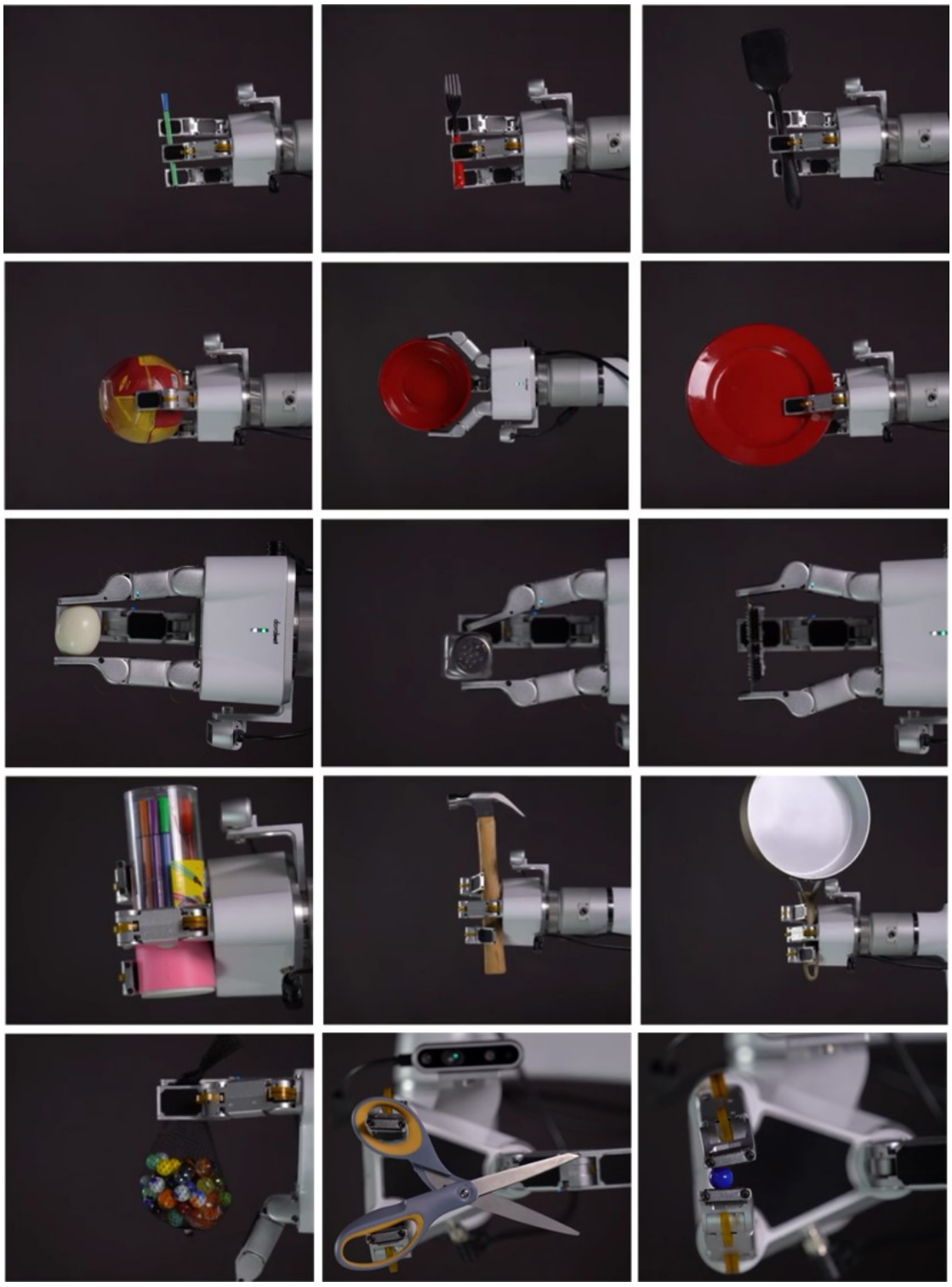


Figure 4.2: Grasp different objects with DoraHand

4.2 Sensing Capability

Sensing capability is a crucial feature of dexterous hands. In addition to the range, sampling rate, and film sensor unit precision, the evaluation of the tactile sensor module capability should also encompass static sensing precision and dynamic sensing capability. This part of the work is presented in two subsections: 4.2.1 experiment design, and 4.2.2 performance.

4.2.1 Experiment Design

To evaluate the static sensing precision, the force distribution on several film sensors inside the module can be used. The experiment is designed to add a force to the sensor module and calculate the force adding position to verify performance. For this purpose, a device must be designed to measure and verify precision. The device includes two motion DOF in the horizontal and vertical directions. The load module controls the vertical direction and can control the payload using feedback from a force sensor. The linear module controls horizontal motion to test the position sensing precision of the sensor module. This device can be used for a single sensor, finger module, or palm module with different fixtures, as shown in Figure 4.3. With this device, the experiment procedures for static sensing can be designed in four steps:

1. Mount the component on the base and move the force tip to the target horizontal position.
2. Add a predefined payload value for a predefined time length, and record sensor data, and position.
3. Unload and move to the next horizontal position with a predefined distance.
4. Repeat the loading procedure and record data.

In the experiment for static sensing, the predefined payload is set at 50% of the sensor range, with a value of 10 N for the finger sensor module. The horizontal moving step is set at 1 mm, and the motion precision can reach 0.01 mm. To minimize the impact of sudden force changes, the payload adding process is slow and the sensor data used for analysis is extracted after the force is stable.

The tests on this device mainly focus on static performance. However, a dynamic capability experiment is also necessary to verify the sensing capability during the grasping process, especially in cases where slippage occurs. This dynamic sensing performance test can be designed in a simple way by grasping an object on a plane and recording the tactile sensor data throughout the test. The change in sensor data can represent the tactile sensing capability in a natural way.

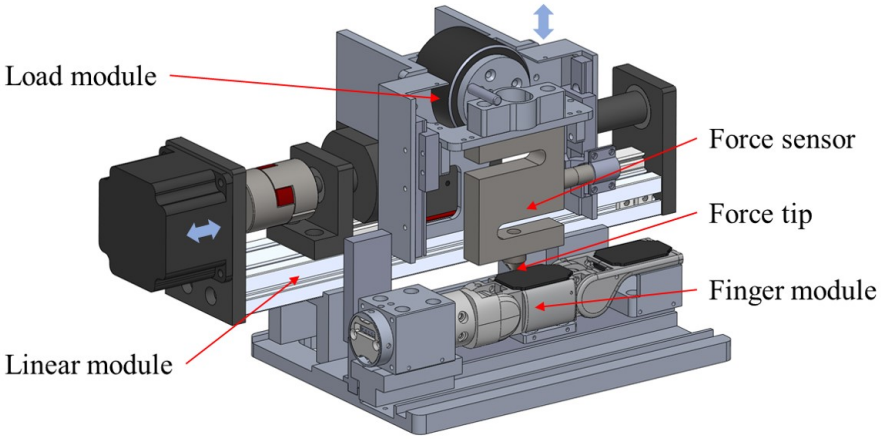


Figure 4.3: Device for sensor calibration and evaluation

4.2.2 Performance

The force sensing range and precision are first evaluated with the device. The force sensing range of a single film sensor is 10 N, the finger knuckle tactile sensor module is 20 N, and the palm tactile sensor module is 40 N. The precision of a single film sensor can reach ± 3%, and creep in 10 minutes is around 1.5%. The precision of one tactile sensor module is around ± 5%, which may be influenced by the precision of the mechanical structure. The minimum sensing force is 100 mN, and the force value changes with 10 mN increments which can help the hand to manipulate soft items and sense the stiffness difference of objects.

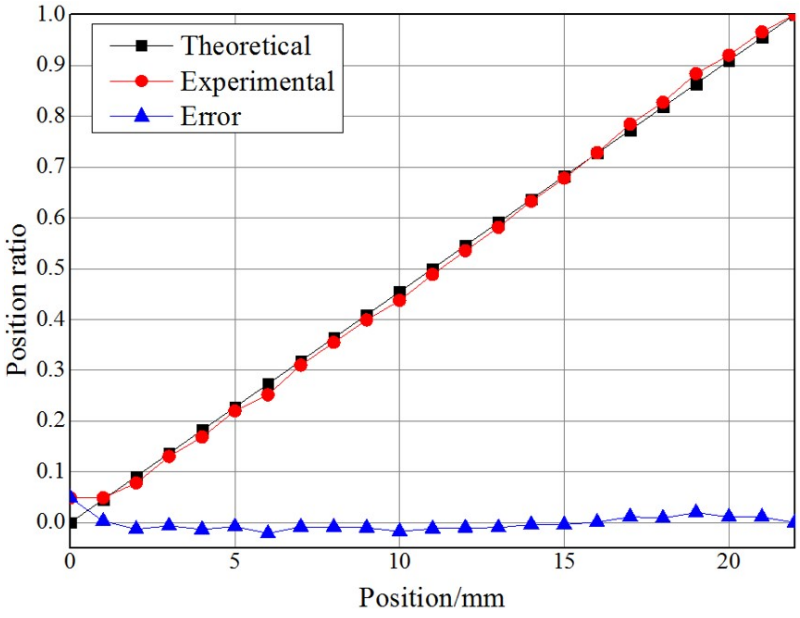


Figure 4.4: Sensor module data with static test

The second part of the experiment focuses on static sensing precision, where the front finger knuckle is selected to test the performance of the tactile sensor module. The test results are shown in Figure 4.4, with a total test distance of 22 mm. The position can be calibrated and can reach a precision with a maximum error of $\pm$ 5%, which is around $\pm$ 1 mm. The most significant error occurred at the edge of the testing area and was caused by differences in contact conditions between the sensor and rubber.

The third part of the experiment focuses on dynamic sensing capability and is shown in Figure 4.5. The force sensing by the sensor module can reach 60 Hz, which meets the requirements for grasp stability evaluation. The force shows a grasping process starting from 0.1 s, with a rapid increase from 0.35 s to 0.45 s, followed by a force balance process from 0.5 s to 1.0 s, with a significant change at 1.0 s. A force balance process that happened in the hand can be detected with such data.

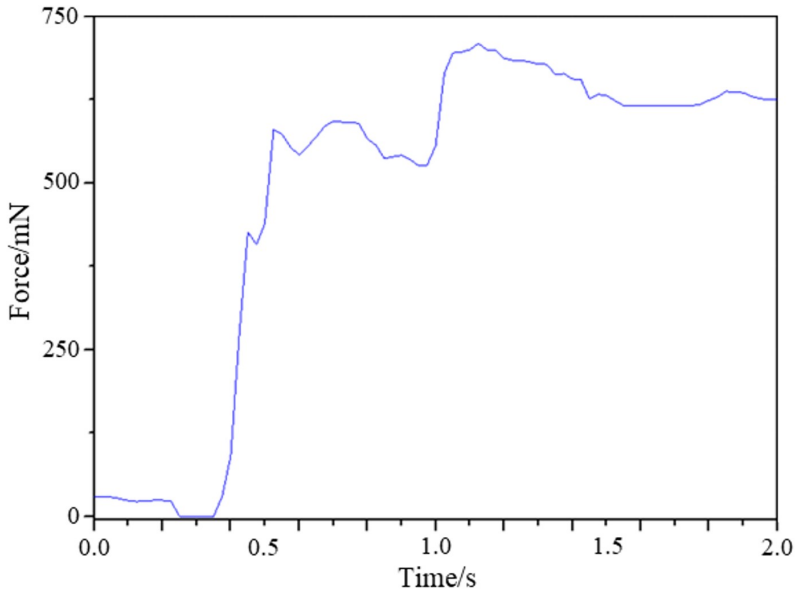


Figure 4.5: Sensor module data with dynamic test

In conclusion, The sensor unit performs well in sensing capability, force point position, and dynamic tactile sensing. The sensor unit design achieves nearly $\pm$ 5% perception accuracy in static experiments and can detect vibration data in dynamic scenarios. With this sensor unit, the robot hand can perform better in tactile sensing and stable grasping research.

4.3 Extension Applications

The dexterous hand capability is evaluated in its three-finger version in previous sections. As a modular dexterous hand with finger modules, the extension capability also needs to be evaluated. Common applications with two and five-finger hands have been designed. In addition to these extensions, researchers can also change the layout and mechanism design to meet different requirements. Two parts of the experiment were designed: 4.3.1 two-finger hand, and 4.3.2 five-finger hand.

4.3.1 Two-Finger Hand

This experiment aims to showcase the extension capability of the finger module and demonstrate the capability of the two-finger hand. To showcase the manipulation capabilities, an actual requirement of the sample pipetting operation experiment has been designed. This operation involves several key actions: picking up a tube, unscrewing the tube cap, operating the pipette, tightening the tube cap, and placing the tube back. The end-effector must be able to manipulate both the tube cap and the pipette. Due to the limited dimensions of the tube, the cap can only be grasped with two fingers, and operating the pipette can be accomplished with one DOF. With these requirements, combining the two-finger and three-finger hands can accomplish the whole procedure. The details of this work are divided into two parts: 4.3.1.1 main hardware design, and 4.3.1.2 application performance.

4.3.1.1 Main Hardware Design

To meet the requirements of sample pipetting operation, several hardware components need to be designed. These include a two-finger hand, a pipette operation device with a three-finger hand, and the overall system layout.

The two-finger hand includes two finger modules, which means the electronic hardware can remain the same as the three-finger hand, with the only difference being the layout of the palm. To increase the flexibility of the palm, two rotational DOF in the three-finger palm are kept. The overall layout of the two-finger hand is similar to that of the three-finger hand, but without the thumb position. Due to the limited space on the palm of the two-finger hand, a finger knuckle tactile sensor module is used to replace the three-finger hand palm. The design of the two-finger hand can be seen in Figure 4.6 (a).

The three-finger hand is used to grasp tubes and operate a pipette. The finger in the thumb position is used to operate the pipette, while the other two fingers are used for grasping tubes. A linkage structure is mounted on the finger in the thumb position and is controlled through the hand's control system. This structure can be

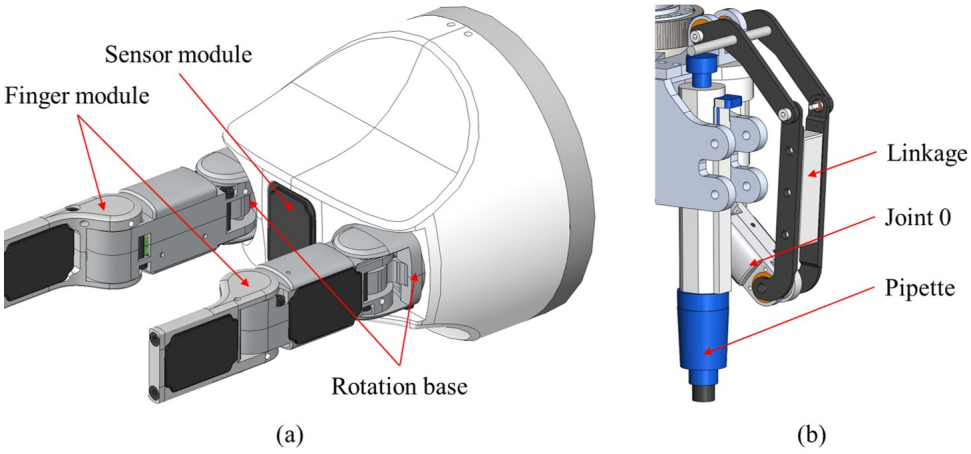


Figure 4.6: Design of two-finger hand and three-finger hand

seen in Figure 4.6 (b).

To loosen the tube cap, a large enough torque is needed, and the friction at the fingertips needs to be increased. All fingertips have been modified and mounted with 3D-printed components. The tip designs for both the two-finger and three-finger hands are slightly different due to their different uses. The designs for this application are shown in Figure 4.7.

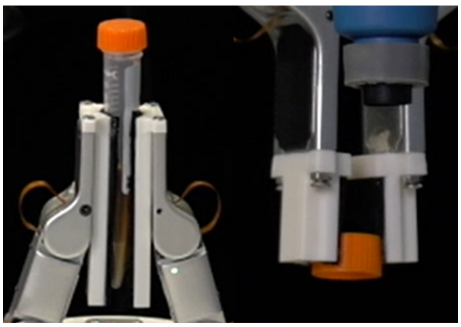


Figure 4.7: Fingertip design for tube operation

The remaining parts include a Universal Robots (UR)5 robot arm, a RealSense D435, two rotational bases for two two-finger hands, a barcode scanner, two tube racks, a pipette tip rack, and other materials for pipette operation. The entire application system can be seen in Figure 4.8.

4.3.1.2 Application Experiment

Referring to a real-world application of sample pipetting, the detailed operation flow of the experiment can be broken down into several steps.

1. Pick up a new tube from the rack and place it in the two-finger hand.

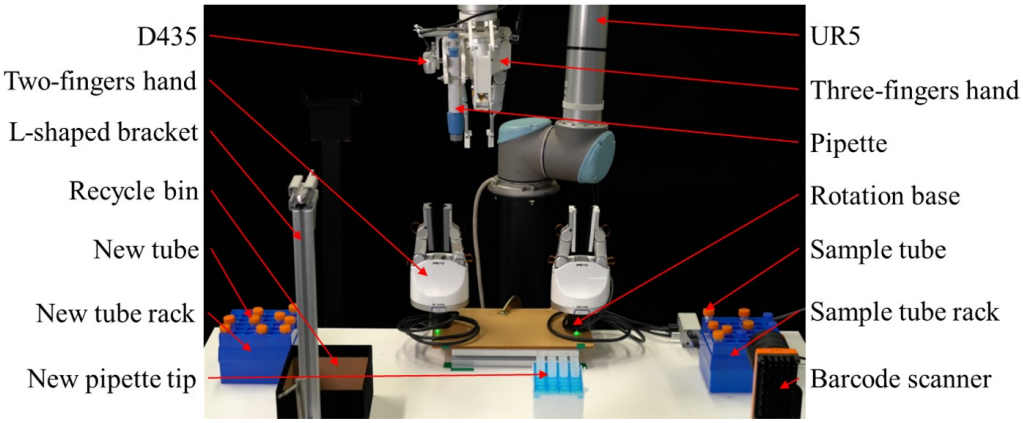


Figure 4.8: The system of sample pipetting application

2. Pick up the sample tube from the rack, scan its barcode, and place it in the two-finger hand.
3. Loosen the cap of the sample tube and pick it up.
4. Get a new pipette tip and absorb the sample from the sample tube.
5. Close the cap of the sample tube.
6. Loosen the cap of a new tube and pick it up.
7. Transfer the sample to the new tube.
8. Close the cap of the new tube.
9. Put the pipette tip in the recycle bin.
10. Move the new tube back to its rack.
11. Move the sample tube back to its rack.

The application is completed with two-finger and three-finger hands. The operation for pipetting is integrated with the hand control and is smooth. Picking up tiny objects like caps is easy with two finger grasping. The biggest challenge during the experiment is operating the tube cap. The force required to loosen the cap must be large enough and the grasp must be stable enough. Otherwise, any change in the cap's position within the fingers causes failure when closing it. This challenge has been addressed with this finger module design. The process of loosening the cap of a new tube is shown in Figure 4.9.

This application demonstrates the capabilities of both two-finger and three-finger hands, including sufficient fingertip force for loosening caps, precise hand operation

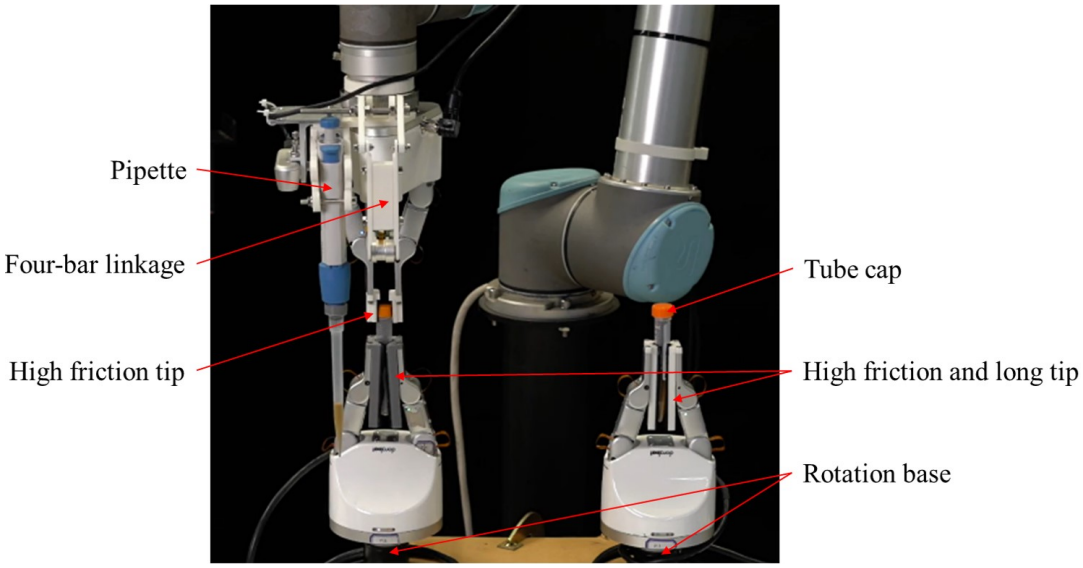


Figure 4.9: Dexterous hand application in pipette application

for handling small objects like tube caps, flexible fingertip extension, and easy redesign of the three-finger hand. Furthermore, one advantage of the two-finger hand is its smaller palm size, which allows it to fit into narrow spaces that require more precise manipulation.

4.3.2 Five-Finger Hand

The five-finger hand extension experiment focuses on its design and basic performance. The five-finger dexterous hand is designed using five finger modules mounted onto a newly designed palm. Referring to the DOF in a human hand, there are twenty-one DOF in the hand. Considering that the four DOF in four DIP joints have been ignored with the finger module, there remain seventeen DOF in the hand, which include ten DOF in the five finger modules. This means that seven additional DOF need to be added to the five-finger palm. Like the human hand, there are four DOF for abduction and adduction at the base of four fingers, which can be simplified using a linkage structure. The remaining three DOF are located at the base of the thumb. Specifically, two DOF have been added to the base of the thumb finger and one DOF has been added to the base of the other four fingers. The design of the five-finger hand can be seen in Figure 4.10.

The sensor design for the five-finger hand is similar to the palm tactile sensor of three-finger hand, but has been modified to fit the shape of the hand. In total, two sensor modules and four motors have been added for the five-finger hand. Due to the increased requirements for motor control and sensor acquisition, the extension function of the main board and pico board is applied in this design, using one main

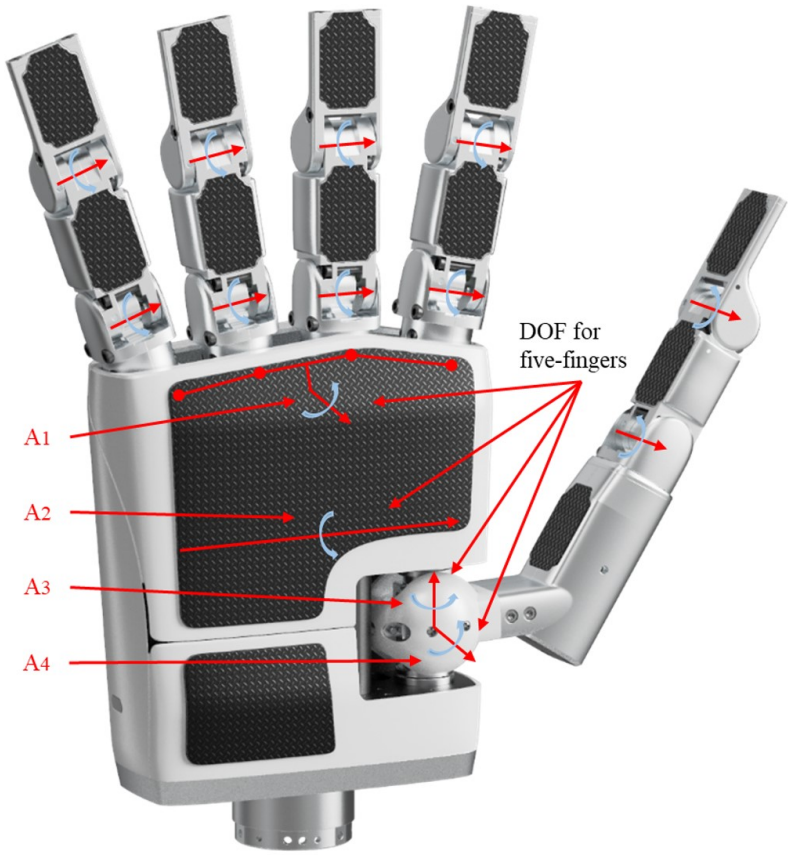


Figure 4.10: Design of five-finger hand

board and two pico boards.

The five-finger hand has a total of seventeen DOF and is driven by fourteen motors. The joint range of each finger is still kept at $\pm$ 90°, with the range of joint A_1 and A_1 being -10° - 0°, the range of joint A_3 being -30° - 90°, and the range of joint A_4 being $\pm$ 30°. The hand weights 2.5 kg. With the same finger module as the three-finger hand, the fingertip grip force is the same value of 25 N. The encompassing grip payload highly depends on the shape of the object. With the same cylinder, it can provide more than 7 kg payload during testing. In addition to finger torque and palm material, the limitation of the five-finger hand's payload is also influenced by the torque and strength of the thumb base. The basic parameters of the five-finger hand are shown in Table 4.3.

Table 4.3: Basic parameters of five-finger hand

Dimension L-W-H	Weight	Finger weight	Joint number	Joint speed & precision
300 mm-220 mm-64 mm	2500 g	180 g	14	70°/s / $\pm$ 0.5°
Sensor module number	Input power	Communication type & rate	Fingertip grip force	Encompassing grip payload
13	24 V / 4 A	Ethernet or USB / 60 Hz	25 N	7 kg

The five-finger hand was tested with various gestures and grasping actions, demonstrating its ability to mimic most human hand gestures. Examples of these tests can be seen in Figure 4.11. The design and usage of the five-finger dexterous hand show the versatility of its modular design, allowing users to customize it according to their specific needs.

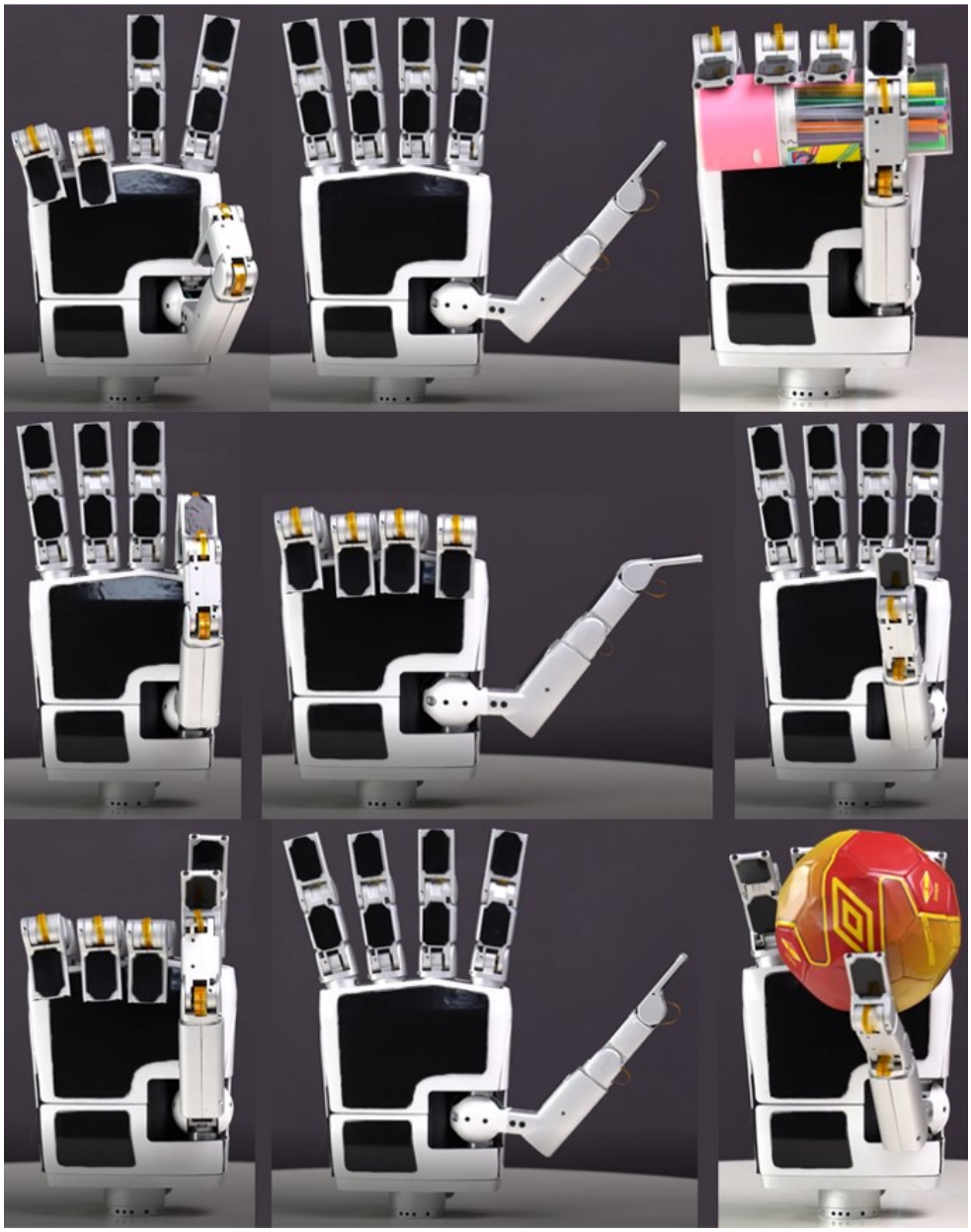


Figure 4.11: Different gestures and grasp poses with five-finger hand

4.4 Conclusion

This chapter provides a comprehensive evaluation of the dexterous hand from three different perspectives, offering valuable insights into effective evaluation methods for the hardware and showcasing its capabilities. This evaluation process serves as a useful guide for advancing research in robotic manipulation and gaining a deeper understanding of the dexterous hand's capabilities and user-friendliness. The results of the evaluation clearly demonstrate the high performance of the dexterous hand. Its basic parameters can meet most requirements when compared to other hands. The integrated tactile sensing module shows good performance and can support tactile-related applications. The finger module has good extension capability, and besides the hot-swap function, the extensions of both two-finger and five-finger hands have been verified. The following part focuses on grasp planning with the DoraHand, specifically its ability to execute grasps using a combination of analytic and data-driven methods.

Part III

Grasp planning

Chapter 5

Analytic Grasp Planning for Three-Finger Dexterous Hand

Grasping is essential for the successful use of a dexterous hand. Despite significant progress in 2D and 3D analytic grasping development, it remains a complex field that requires consideration of various conditions. To better demonstrate the capabilities of the dexterous hand and provide a reference, it can be helpful to focus on a common and simplified scene. This thesis project focuses on grasp planning within the Table-Top scene, with targets simplified as 2D polygons from a top view.

This chapter introduces analytic grasp planning for the three-finger dexterous hand DoraHand. It is divided into four sections. Section 5.1 provides a basic analysis for grasp planning. Section 5.2 presents the grasp plan with the limitations of the dexterous hand. Section 5.3 verifies the algorithm with actual calculations. Section 5.4 provides a summary of this chapter.

5.1 Analytic Analysis

Previous research has extensively studied analytic grasp planning [Li et al., 2002, Zhang et al., 2022]. However, to accommodate the specific hand layout and limitations of the three-finger hand in this thesis, additional work is required. This section is divided into two subsections: 5.1.1 grasp plan with force-closure, and 5.1.2 grasp plan with quality evaluation.

5.1.1 Grasp Plan with Force-Closure

For grasping, form-closure is a stable status that is difficult to achieve and limited by the dimension and shape of the target. Force-closure, on the other hand, is an achievable status for a dexterous hand. The grasp plan with force-closure will be delivered

in this work. To achieve a force-closure grasp plan, force and moment equilibrium must be simultaneously achieved. This subsection focuses on the 2D situation within the target Table-Top scene. Before the analysis, certain assumptions must be made to define the target scene.

1. The object is a rigid body with precise geometry and evenly distributed mass;
2. A Coulomb static friction model defines the friction cone, with a constant coefficient μ;
3. The touch point and force can be applied perfectly;

If the grasp reaches force-closure in 2D space, the force and moment in the GWS should have a value of $\vec{0}$ as shown in Equation 5.1.

$$W = \begin{bmatrix} \vec{F} \\ \vec{M} \end{bmatrix} = \begin{bmatrix} \vec{0} \\ \vec{0} \end{bmatrix} \tag{5.1}$$

According to the definition of Coulomb friction, a force $\vec{f_i}$ is applied at point C_i, and there exists a friction cone defined by two unit vectors $\vec{e_{i1}}$ and $\vec{e_{i2}}$. The force vector $\vec{f_i}$ can be represented with two parameters, k_{i1} and k_{i2}, as shown in Equation 5.2. The directions of the unit vectors $\vec{e_{i1}}$ and $\vec{e_{i2}}$ are determined by the contact surface normal vector $\vec{n_c}$ and friction coefficient μ, and the angle between them is 2α, where $\alpha = arctan(\mu)$. With the vector from the point O to the contact position as $\vec{r_{oi}}$, the moment produced by the external force can be calculated as $\vec{m_{oi}} = \vec{r_{oi}} \times \vec{f_i}$. Figure 5.1 illustrates the relationship between the force and moment at the contact point C_i and point O.

$$\vec{f_i} = k_{i1}\vec{e_{i1}} + k_{i2}\vec{e_{i2}},\ k_{il}, k_{i2} \geq 0 \tag{5.2}$$

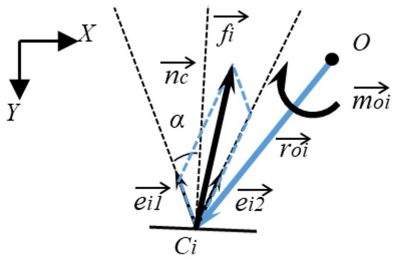


Figure 5.1: Coulomb friction, force, and moment

By combining the force and moment equations with the GWS equation, the force and moment equilibrium situation with n fingers can be presented in equations 5.3, 5.4, 5.5.

$$\vec{F} = \sum_{i=1}^{n} \vec{f_i} = \sum_{i=1}^{n} (k_{i1}\vec{e_{i1}} + k_{i2}\vec{e_{i2}}) = \vec{0},\ k_{il}, k_{i2} \geq 0 \tag{5.3}$$

$$\vec{M} = \sum_{i=1}^{n} \vec{m_{oi}} = \sum_{i=1}^{n} \vec{r_{oi}} \times \vec{f_i} = \sum_{i=1}^{n} \vec{r_{oi}} \times (k_{i1}\vec{e_{i1}} + k_{i2}\vec{e_{i2}}) = \vec{0},\ k_{il}, k_{i2} \geq 0 \tag{5.4}$$

$$W = \begin{bmatrix} \vec{F} \\ \vec{M} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} (k_{i1}\vec{e_{i1}} + k_{i2}\vec{e_{i2}}) \\ \sum_{i=1}^{n} \vec{r_{oi}} \times (k_{i1}\vec{e_{i1}} + k_{i2}\vec{e_{i2}}) \end{bmatrix} = \begin{bmatrix} \vec{0} \\ \vec{0} \end{bmatrix} \tag{5.5}$$

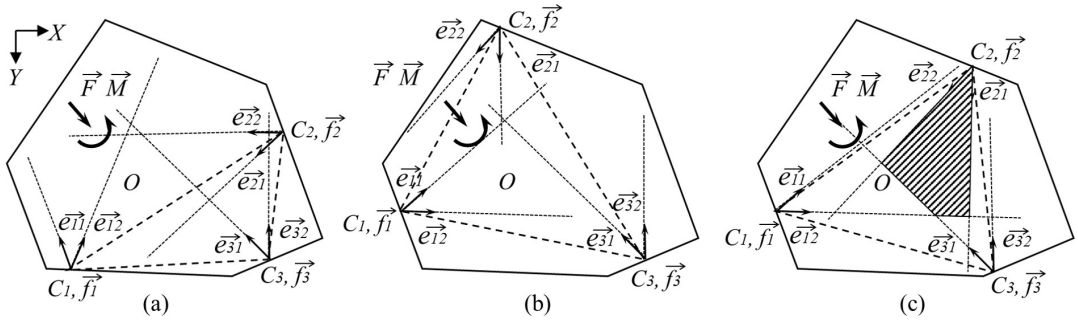


Figure 5.2: Frictional three-finger 2D grasp

For the development of the algorithm to verify the force-closure situation shown in these equations, some assumptions and changes to the conditions are needed. Based on the previous work on analytic approaches, many propositions have been well studied and verified. For a 2D grasp with three fingers, parallel grasping is a special case that can be simplified as two fingers grasping in the force and moment equilibrium part. Therefore, the following discussion only considers non-parallel grasping cases. Proposition 1 provides a statement about force-closure: nonmarginal equilibrium can achieve force-closure [Ponce and Faverjon, 1995]. When considering force and moment equilibrium, there are three typical cases, as illustrated in Figure 5.2. In case (a), it is clear that these three forces cannot intersect at one point, and the moment cannot achieve equilibrium. Case (b) can reach force and moment equilibrium when the three force directions and values are fixed. This situation is marginal equilibrium and cannot meet the requirement of force-closure. Case (c) is the most stable one, and there is a range for the values of k_{ij}, $i = 1, 2, 3$, $j = 1, 2$. This situation is a force-closure with nonmarginal equilibrium.

Proposition 1. *A 2-D, three-finger grasp that achieves nonmarginal equilibrium also achieves force closure.*

When force-closure is achieved with nonmarginal equilibrium, moment equilibrium should be reached at any reference point. By selecting one point inside C_i,

$i \in 1, 2, 3$ as the coordinate point and choosing $i = 2$, the moment calculation on point C_2 can be simplified as shown in Equation 5.6 and should be $\vec{0}$. The scheme is shown in Figure 5.3. In a force-closure case, for the $i \in 1, 2, 3$, there should be $\vec{M}_i = \vec{0}$.

$$\vec{M} = \vec{M}_i = \vec{m_{21}} + \vec{m_{23}} = \vec{C_2C_1} \times (k_{11}\vec{e_{11}} + k_{12}\vec{e_{12}}) + \vec{C_2C_3} \times (k_{31}\vec{e_{31}} + k_{32}\vec{e_{32}}) = \vec{0} \quad (5.6)$$

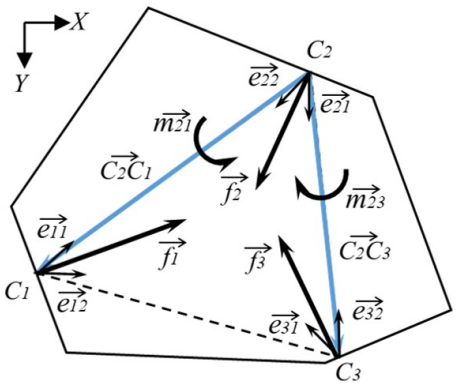


Figure 5.3: Moment equilibrium at point C_2

Due to moment equilibrium, $\vec{m_{ij}}$ and $\vec{m_{ik}}$ should have opposite signs. Then it can be represented as $\vec{m_{ij}} \cdot \vec{m_{ik}} < 0$. Since the value is influenced by many factors, the sign of each part can help to calculate more easily. Two sign functions are defined for the equilibrium calculation. The first sign function is called $Sgn\_3(x)$, whose definition is shown in Equation 5.7. It can be used for the judgment of the component moment at each contact point, $Sgn_{mij} = Sgn\_3(\vec{m_{ij1}}) + Sgn\_3(\vec{m_{ij2}})$, where $\vec{m_{ij1}} = \vec{C_iC_j} \times \vec{e_{j1}}$ and $\vec{m_{ij2}} = \vec{C_iC_j} \times \vec{e_{j2}}$. The second sign function is called $Sgn\_2(x)$, whose definition is shown in Equation 5.8. It can be used for the judgment of the combined moment at point C_i, $Sgn_{Mi} = Sgn\_2(Sgn_{mij} \cdot Sgn_{mik})$. If $Sgn_{Mi} == -1$, it means that it is impossible to reach moment equilibrium at point C_i. If $Sgn_{Mi} == 1$, it means that it is possible to reach moment equilibrium and meet the force-closure requirement at point C_i. The algorithm for moment equilibrium judgment can be checked in Algorithm 1.

$$Sgn\_3(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \quad (5.7)$$

$$Sgn\_2(x) = \begin{cases} 1, & x \leq 0 \\ -1, & x > 0 \end{cases} \quad (5.8)$$

Algorithm 1 Moment equilibrium judgment

function MOMENT EQUILIBRIUM JUDGMENT FUNCTION($points$)
  Calculate $\vec{r_{ij}}, \vec{e_{ik}}, i, j = 1, 2, 3, k = 1, 2$ from $points$
  $n = 0$
  for $i = 1, 2, 3$ **do** ▷ Check moment equilibrium at point C_i
    for $j = 1, 2, 3$, $j \neq i$ **do**
      for $k = 1, 2$ **do**
        $\vec{m_{ijk}} = \vec{r_{ij}} \times \vec{e_{jk}}$
      end for
      $Sgn_{mij} = Sgn\_3(\vec{m_{ij1}}) + Sgn\_3(\vec{m_{ij2}})$
    end for
    $Sgn_{Mi} = Sgn\_2(Sgn_{mij} \cdot Sgn_{mik}), i \neq j \neq k$
    if $Sgn_{Mi} == 1$ **then**
      $n = n + 1$
    end if
  end for
  if $n == 3$ **then**
    return $True$
  else
    return $False$
  end if
end function

The judgment provided by the Algorithm 1 can help to quickly judge the situation, but it cannot guarantee that force and moment equilibrium are achieved at the same time and cannot prove that the equilibrium is nonmarginal equilibrium. A sufficient condition for the force-closure situation is still needed. Proposition 2 can meet this requirement and help to confirm the interaction zone of the contact point force [Ponce and Faverjon, 1995].

Proposition 2. *A necessary and sufficient condition for the existence of three nonzero contact forces, not all of them being parallel, which achieve equilibrium is that there exist three forces in the friction cones at contact points which positively span the plane and whose lines of action intersect at some point.*

For the positively span, it can be achieved when the three forces meet the requirement of $Sgn_{Mi} == 1$ for $i = 1, 2, 3$. If $\vec{m_{ij}} > 0$, there should be $\vec{m_{ik}} < 0$, and the direction of $\vec{f_k}$ should be located between $\vec{C_kC_i}$ and $\vec{e_{k2}}$. Otherwise, $\vec{m_{ik}} > 0$ and equilibrium cannot be reached. This situation means that $\vec{e_{k1}}$ in the $\vec{f_k} = k_{k1}\vec{e_{k1}} + k_{k2}\vec{e_{k2}}$ can be replaced by $\vec{C_kC_i}$ and form a new function of $\vec{f_k} = c_{k1}\vec{C_kC_i} + k_{k2}\vec{e_{k2}}$. This change can make the calculation of c_{k1} and k_{k2} locate in the available range and help to reduce some calculations. For this consideration, when there is a potential case that can meet equilibrium, it can check the e_{k1} and e_{k2} for whether the vector is located between the line of C_iC_k and C_jCk. Set the C_i, C_j , C_k, e_{k1}, e_{k2} in a clockwise direction,

define the check and replace equation as Equation 5.9, there can get a new disposition H that can help to calculate the result faster. Based on the Proposition 2, combined with the updated disposition H, there can get the Proposition 3 [Li et al., 2003].

$$\begin{cases} \text{If } Sgn\_3(C_i\vec{C}_k \times e\vec{_{k1}}) * Sgn\_3(C_j\vec{C}_k \times e\vec{_{k1}}) > 0 \\ e\vec{_{k1}} = \begin{cases} C_k\vec{C}_i / |C_iC_k| & \text{If } Sgn\_3(C_i\vec{C}_k \times e\vec{_{k1}}) > 0 \text{ and } Sgn\_3(C_j\vec{C}_k \times e\vec{_{k2}}) < 0 \\ C_i\vec{C}_k / |C_iC_k| & \text{If } Sgn\_3(C_i\vec{C}_k \times e\vec{_{k1}}) < 0 \text{ and } Sgn\_3(C_j\vec{C}_k \times e\vec{_{k2}}) > 0 \\ \vec{0} & \text{Others} \end{cases} \\ \text{If } Sgn\_3(C_i\vec{C}_k \times e\vec{_{k2}}) * Sgn\_3(C_j\vec{C}_k \times e\vec{_{k2}}) > 0 \\ e\vec{_{k2}} = \begin{cases} C_k\vec{C}_j / |C_jC_k| & \text{If } Sgn\_3(C_j\vec{C}_k \times e\vec{_{k2}}) < 0 \text{ and } Sgn\_3(C_i\vec{C}_k \times e\vec{_{k1}}) > 0 \\ C_j\vec{C}_k / |C_jC_k| & \text{If } Sgn\_3(C_j\vec{C}_k \times e\vec{_{k2}}) > 0 \text{ and } Sgn\_3(C_i\vec{C}_k \times e\vec{_{k1}}) < 0 \\ \vec{0} & \text{Others} \end{cases} \end{cases} \tag{5.9}$$

Proposition 3. *A necessary and sufficient condition for the three-finger equilibrium grasp is that the intersection of the three double-side friction cones is not empty while the disposition H is done.*

With the disposition H and the Proposition 3, the remaining challenge is to solve for the intersection points. When there are intersection points within the friction cones, it means that the intersection points of two friction cones should be located inside the third friction cone. Otherwise, there will be no intersection points among the three friction cones.

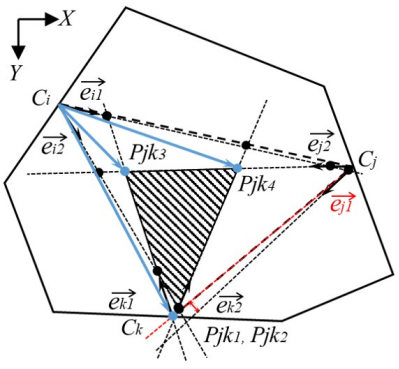


Figure 5.4: Twelve intersection points of friction cones

To calculate the interaction points of two friction cones at point C_j and C_k, the points of P_{jkn}, $n = 1, 2, 3, 4$ can be obtained. If a point is locate inside the friction cone of C_i, there should be $Sgn\_3(C_i\vec{P}_{jkn} \times e\vec{_{i1}}) \cdot Sgn\_3(C_i\vec{P}_{jkn} \times e\vec{_{i2}}) <= 0$. With the requirement of not empty and considering the marginal equilibrium, there should

at least three of these twelve points meet the equation. The points that are inside corresponding friction cones form a list of points P. When the requirement of not empty is met, the force-closure status can be achieved. The non-empty zone can be used for the next step of quality evaluation. The scheme can be seen in Figure 5.4, where $\vec{e_{j1}}$ has been changed according to the disposition H, and the P_{jk1} and P_{jk2} located in the same position. The algorithm for the friction cone intersection points judgment is presented in Algorithm 2.

Algorithm 2 Friction cone intersection points judgment

function FRICTION CONE INTERSECTION POINTS JUDGMENT FUNCTION($points$)
  for $i = 1, 2, 3$ **do** ▷ Check the intersection point location
    $j = (i + 1)\%3$
    $k = (j + 1)\%3$
    for $n = 1, 2, 3, 4$ **do**
      if $Sgn\_3(C_i\vec{P}_{jkn} \times \vec{e_{i1}}) \cdot Sgn\_3(C_i\vec{P}_{jkn} \times \vec{e_{i2}}) <= 0$ **then**
        if $P_{jkn} \neq points$ in P **then**
          $P.append(P_{jkn})$
        end if
      end if
    end for
  end for
  if $len(P) \geq 3$ **then**
    return $True$
  else
    return $False$
  end if
end function

5.1.2 Grasp Plan with Quality Evaluation

In Proposition 3, when a non-parallel grasp plan is force-closure, the three force friction cones can intersect and form a convex cone. The points inside this cone reach force-closure. The size of the convex cone is influenced by the friction cone angle and direction. A larger convex cone allows for a greater range of adjustment in force direction and value, enabling the grasp to adapt to different materials more effectively.

During the grasping process, two main points need to be considered: the margin on the grasp condition and the margin on the grasp position. The first point is related to the fact that the contact status and friction parameters may differ from the values used for calculation, and these differences can have a significant impact on the result. The second point is related to the error of detection and execution, the grasp position needs a suitable margin that can guarantee that the actual grasping also meets the force-closure. All of these factors can be considered as finding a larger available range

for each contact point force.

In reality, the friction coefficient is often unknown, and the real force may not entirely follow the Coulomb static friction model. Therefore, the friction coefficient μ is defined as a constant value in the calculations. Since the area value is positively correlated with the friction coefficient μ, comparing the margin can still indicate the quality of the grasp plan.

For the comparison of the margin, there are three parameters related to the intersection area that can be used for comparison. Area A_c is the interaction area of the three contact points. Area A_n is the area of the interaction convex cone. Radius r_n is the largest inscribed circle radius of the interaction convex cone. The value of A_c has no relationship with the direction of force, it shows the capability of bearing an external torque and can also be used for evaluating parallel grasp plan, but it cannot guarantee force-closure situations. On the other hand, the value of A_n can indicate the flexibility of the grasp, a larger area can be more stable, and the value means it already meets the force-closure requirement. The r_n can better represent the stability when considering all three points together. The best grasp plan passes through the center of the circle, ensuring that position and force have the greatest stability range. The quality evaluation algorithm can be determined by sorting parameters in the following sequence: r_n, A_n, and A_c. Figure 5.5 illustrates the difference among these three criteria.

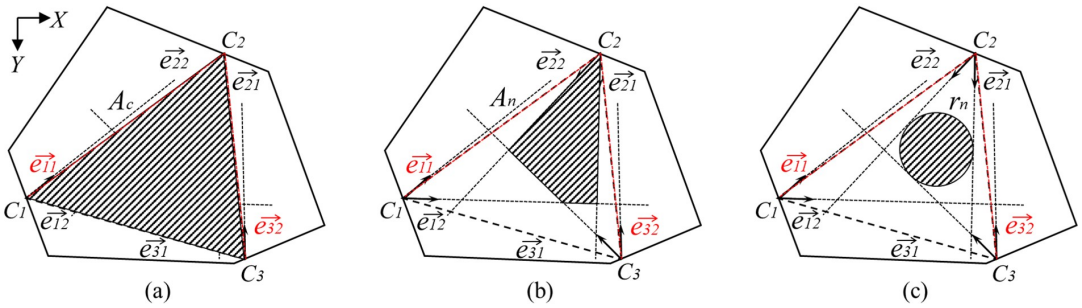


Figure 5.5: Three criteria for grasp quality evaluation

Compare to non-parallel grasping, equilibrium situations with parallel grasping are easier to calculate and evaluate. The A_c can be used in parallel grasping and evaluated, the larger area A_c, the greater grasp stability. In the specific situation for DoraHand, because the distance between finger 1 and finger 2 cannot be changed, the grasp plan can be evaluated with the distance between two edges and the dimension of the target object.

5.2 Grasp Planning with Hand

In addition to evaluating the quality of the grasp plan, another important consideration is the layout of the hand and the range of its joint. All grasp plans must meet the constraints of hand layout and joint range. The grasp plan, including the hand joint coordinate, is shown in Figure 5.6.

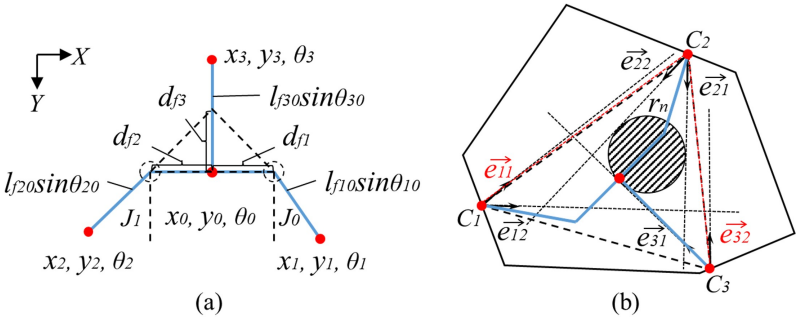


Figure 5.6: Coordinate definition in grasp plan

In the Table-Top scene, the grasp gesture used is limited. To increase stability, the front tip is set in a vertical direction, meaning that the joint angles of the two finger joints are negative to each other. This can help simplify the equation by using one finger joint angle. The position of the fingertips can be described using the joint value with Equations 5.10 - 5.12.

$$\begin{cases} x_1 = x_0 + d_{f1}cos(\theta_0 - 90°) + l_{f10}sin(\theta_{10})cos(\theta_0 - \theta_{00}) \\ y_1 = y_0 + d_{f1}sin(\theta_0 - 90°) + l_{f10}sin(\theta_{10})sin(\theta_0 - \theta_{00}) \\ \theta_1 = \theta_0 - \theta_{00} - 180° \end{cases} \tag{5.10}$$

$$\begin{cases} x_2 = x_0 + d_{f2}cos(\theta_0 + 90°) + l_{f20}sin(\theta_{20})cos(\theta_0 + \theta_{01}) \\ y_2 = y_0 + d_{f2}sin(\theta_0 + 90°) + l_{f20}sin(\theta_{20})sin(\theta_0 + \theta_{01}) \\ \theta_2 = \theta_0 + \theta_{01} - 180° \end{cases} \tag{5.11}$$

$$\begin{cases} x_3 = x_0 + d_{f3}cos(\theta_0 - 180°) + l_{f30}sin(\theta_{30})cos(\theta_0 - 180°) \\ y_3 = y_0 + d_{f3}sin(\theta_0 - 180°) + l_{f30}sin(\theta_{30})sin(\theta_0 - 180°) \\ \theta_3 = \theta_0 \end{cases} \tag{5.12}$$

In this context, x_i, y_i, and θ_i, where $i \in 1, 2, 3$, represent the position parameters of fingertip i in the grasp plan. x_0, y_0 represent the position of the coordinate center of the palm, d_{fi}, $i \in 1, 2, 3$, represents the position parameter of the finger base, l_{fij}, $i \in 1, 2, 3$, $j \in 0, 1$, represents the length of each finger i joint j. θ_{ij}, $i \in 0, 1, 2, 3$, $j \in 0, 1$ represents the value of joint j for each part i. The complete

position and joint command for this dexterous hand and arm are given by vector $[x_0, y_0, \theta_0, \theta_{00}, \theta_{01}, \theta_{10}, \theta_{20}, \theta_{30}]$. The remaining commands are fixed under the condition of Table-Top grasping and grasping an object with vertical fingertips.

Once the contact point locations C_i, $i = 1, 2, 3$, are determined, the normal direction of the contact plane can be obtained. From this, the grasp plan can be determined in the data format of $[x_{ci}, y_{ci}, \theta_{ci}]$, where $i = 1, 2, 3$. The calculation process is divided into parallel and non-parallel cases. The non-parallel case parameter of the dexterous hand can be obtained using Equation 5.13. When the $\theta_{00} = 0°$, and $\theta_{01} \neq 0°$, the equation need to be changed to calculate the angle on the non-zero angle side first.

$$\begin{cases} \theta_0 = \theta_{c3} \\ \theta_{00} = \theta_0 - \theta_{c1} - 180° \\ \theta_{01} = \theta_{c2} - \theta_0 + 180° \\ \theta_{10} = arcsin(\dfrac{(x_{c1} - x_{c3})sin(\theta_0) - (y_{c1} - y_{c3})cos(\theta_0) - d_{f1}}{l_{f10}sin(\theta_{00})}),\ \theta_{00} \neq 0° \\ x_0 = x_{c1} - d_{f1}sin(\theta_0) + l_{f10}sin(\theta_{10})cos(\theta_{c1}) \\ y_0 = y_{c1} + d_{f1}cos(\theta_0) + l_{f10}sin(\theta_{10})sin(\theta_{c1}) \\ \theta_{20} = arcsin(\dfrac{\sqrt{(x_{c2} - x_0 + d_{f2}sin(\theta_0))^2 + (y_{c_2} - y_0 - d_{f2}cos(\theta_0))^2)}}{l_{f20}}) \\ \theta_{30} = arcsin(\dfrac{\sqrt{(x_{c3} - x_0 + d_{f3}cos(\theta_0))^2 + (y_{c_3} - y_0 + d_{f3}sin(\theta_0))^2)}}{l_{f30}}) \end{cases} \tag{5.13}$$

For the parallel grasping case with three fingers, both θ_{00} and θ_{01} are equal to $0°$. The joint value of three fingers is set to the same value to achieve symmetrical grasping. The calculation result of parallel grasp is shown in Equation 5.14.

$$\begin{cases} \theta_0 = \theta_{c3} \\ \theta_{00} = 0° \\ \theta_{01} = 0° \\ \theta_{10} = \\ arcsin(\dfrac{\sqrt{(x_{c1} - x_{c3} - d_{f10}sin(\theta_0) - d_{f30}cos(\theta_0))^2 + (y_{c1} - y_{c3} + d_{f10}cos(\theta_0) - d_{f30}sin(\theta_0))^2}}{(l_{f10} + l_{f30})^2}) \\ x_0 = x_{c1} - d_{f1}sin(\theta_0) + l_{f10}sin(\theta_{10})cos(\theta_{c1}) \\ y_0 = y_{c1} + d_{f1}cos(\theta_0) + l_{f10}sin(\theta_{10})sin(\theta_{c1}) \\ \theta_{20} = \theta_{10} \\ \theta_{30} = \theta_{10} \end{cases} \tag{5.14}$$

Additionally, it is important to note that the joint limitations for each finger joint are within $[-90°, 90°]$. These limitations must be considered during grasp planning. If the grasp plan does not meet these requirements, it cannot be used with the dexterous hand. Furthermore, when the hand moves in 3D space, the coordinates should be transformed using the transformation matrix. The entire process, from grasp planning to determining palm position and joint value, can be accomplished using Algorithm 3.

Algorithm 3 Grasp parameters calculation

function GRASP PARAMETERS CALCULATION FUNCTION($points$)
  for $i = 1, 2, 3$ **do**
    $j = (i + 1)\%3$
    $k = (j + 1)\%3$
    if $\theta_{ci} == \theta_{cj}$ and $abs(\theta_{ci} - \theta_{ck}) == 180°$ **then** ▷ Parallel grasping case
      $Joint\_value = [x_0, y_0, \theta_0, 0°, 0°, \theta_{10}, \theta_{20}, \theta_{30}]$
      if $Joint\_value$ in the range $[-90°, 90°]$ **then**
        return $Joint\_value$
      else
        return NULL
      end if
    else ▷ Non-parallel grasping case
      $Joint\_value = [x_0, y_0, \theta_0, \theta_{00}, \theta_{01}, \theta_{10}, \theta_{20}, \theta_{30}]$
      if $Joint\_value$ in the range $[-90°, 90°]$ **then**
        return $Joint\_value$
      else
        return NULL
      end if
    end if
  end for
end function

5.3 Verification of the Analytic Algorithm

The previous section outlines an analysis and calculation process for obtaining a grasp plan with a 2D input. An algorithm is necessary to describe and execute the process which can be divided into three main parts. The first part involves generating the grasp plan with three points and their corresponding direction. This can be accomplished in two steps: selecting the contact edges and forming the edge list, then generating the point using a specific percentage of each edge vector. The second part involves verifying the result with the dexterous hand by sequentially applying Algorithm 1 and 2, and calculating the r_c, A_c, and A_n. The third part involves sorting the grasp plans based on r_c, A_c, and A_n, and selecting the best grasp plan to output

the joint values. These values can then be obtained using Algorithm 3. The complete procedure is illustrated in Algorithm 4.

Algorithm 4 Analytic grasp plan

function ANALYTIC GRASP PLAN FUNCTION($edges$)
  for i in $len(edges)$ **do** ▷ Select the contact edges
    for j in $len(edges)$, $j \neq i$ **do**
      for k in $len(edges)$, $k \neq j \neq i$ **do**
        Push $[i, j, k]$ into $edge\_list$
      end for
    end for
  end for
  for $edge$ in $edge\_list$ **do** ▷ Generate grasp points on the edge
    for i in num **do**
      for j in num **do**
        for k in num **do**
          $point[0] = \vec{edge}[0] * i/num$
          $point[1] = \vec{edge}[1] * j/num$
          $point[2] = \vec{edge}[2] * k/num$
          Push $point$ into $point\_list$
        end for
      end for
    end for
  end for
  for $point$ in $point\_list$ **do** ▷ Check force-closure and calculate the parameter
    if Moment equilibrium judgment($point$) **then**
      if Friction cone intersection points judgment($point$) **then**
        Calculate r_n, A_n, A_c with $point$
        Push $[point, r_n, A_n, A_c]$ to $plan\_list$
      end if
    end if
  end for
  Sort $plan\_list$ with r_n, A_n, A_c ▷ Sequence grasp plan and output joint value
  for $plan$ in $plan\_list$ **do**
    if Grasp parameters calculation($point$) **then**
      Return $plan$, $Joint\_value$
    end if
  end for
end function

To validate the algorithm, a polygon was selected and the DoraHand parameters were used to generate grasp plans. These plans were then sorted based on their r_n values and the best grasp plan was identified. The friction coefficient is a critical factor that influences the number of grasp plans. Since it is difficult to determine the actual coefficient and ensure consistency, it can only be decided using experimental

data. In real applications, a list can be used to assign different values to the target using object detection results. In this experiment, a common value of $\mu = 0.55$ was used for the friction coefficient.

For the verification calculation, a polygon with the points $[(6, 17), (33, 6), (56, 8), (62, 31), (45, 54), (18, 37)]$ and a unit of 1 mm was used. Three edges were selected out of the six edges and each edge was divided into $num = 10$ equal parts. Considering the sequence of three edges, there are a total of 60,000 grasp plans. After the calculation and evaluation, 11,514 plans met the force-closure requirement. There were 9,446 plans with r_n larger than 1 mm, 6,587 plans with r_n larger than 2 mm, and 1,526 plans with r_n larger than 5 mm. The largest r_n is 9.29 mm.

Figure 5.7 illustrates five sample grasp plans, including plans with r_n of 9.29 mm, 5 mm, 2 mm, 1 mm, and the minimum of 0.001 mm. In (a), the grasp plan has a larger r_n, A_n, and A_c are large and have similar values. For (b)-(d), the A_n is much smaller than the A_c. Considering the detection error and actual physical parameters, a position difference with 2 mm is not enough for a stable grasp. The (e) is an extreme case close to marginal equilibrium, the intersection zone includes several results but the actual margin is limited. The experimental results demonstrate the fundamental capabilities of the grasp planning algorithm and confirm its ability to select the best grasp plan while considering force-closure and hand layout. Combined with the vision process algorithm, it can be used for grasp planning.

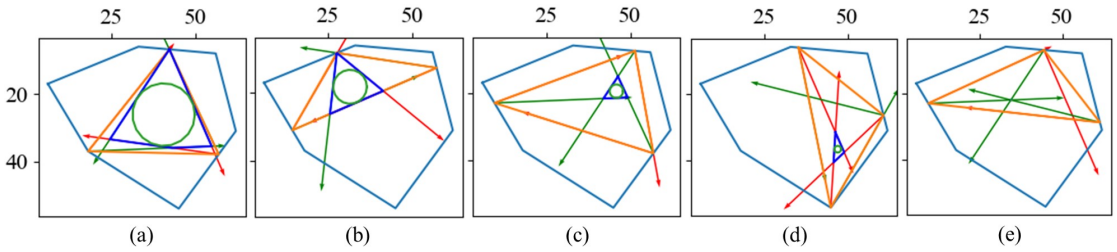


Figure 5.7: Grasp plans with different r_n

5.4 Conclusion

This chapter presents an analytical analysis and algorithm for generating grasp plans in a Table-Top scene using a three-finger dexterous hand. The results demonstrate that the analytic analysis can provide effective solutions with sufficient input parameters and calculations, and can output the optimal grasp plan that meets the force-closure requirement. The radius r_n, area A_n, and area A_c can be used as evaluation criteria and have been verified. However, there are some limitations to the

analytic solution, including the complexity of the calculations and the potential for increased complexity with more complex target shapes. Additionally, the visual process for acquiring the target item shape may have variance at the edges of the target, making the analysis more complex and time-consuming in real-world applications. Inspired by the visual process, a trial with a learning method for generating grasp plans may provide an easy-to-use solution. The next chapter introduces a visual-based learning approach for grasp planning.

Chapter 6

Visual-Based Grasp Planning

Visual data is the fundamental data in grasp tasks. By integrating visual processing with grasp planning, the planning process can be simplified compared to using visual processing alone. As a visual processing and data-driven approach, deep learning has experienced a boom and has been applied in diverse applications. It can effectively analyze the complex information contained in the image. However, data-driven approaches for grasp planning often face challenges such as insufficient data and lower success rates compared to purely visual applications. The diversity of robot grasping tasks and the difficulty in gathering data have hindered progress in this area.

This chapter presents a visual-based grasp planning approach that combines widely used deep learning-based visual techniques with analytic approaches to achieve better results. The target scene remains the same as in the previous section: the Table-Top scene. The chapter is divided into four sections. Section 6.1 introduces the basic concept of the approach. Section 6.2 describes the framework and key components of the model. Section 6.3 shows the performance of the grasp planning. Section 6.4 provides a summary of this chapter.

6.1 Design Concept

In the field of grasp planning, analytic solutions have been developed for a long time, and issues with multiple definitions can be resolved using appropriate methods. However, there remains a gap between theory and practice. The ability to adapt to complex scenes may make the computation complex. A solution that incorporates deep learning capabilities may help with the sampling process and generate more diverse plans. As shown in Figure 1.3, the grasp planning work is in the first stage of pre-grasp manipulation, and can be divided into three steps: 1) gathering data about the target item; 2) proposing available grasp poses based on existing data; 3) excluding some solutions and selecting the best one according to the environment. Combined

with an analytic approach, a deep learning-based grasp planning approach can cover all three of these steps.

For step 1, precise object detection is crucial in the grasping process. It helps to accurately segment and classify objects and retrieve information from past experiences. This can be achieved through a neural network trained on relevant data.

For step 2, a grasp plan can be proposed by a model. The model's ability to generalize can help it be applied to more complex scenes that are difficult to generate using traditional methods. An analytic solution can also provide evaluation results for training the network.

For step 3, the grasp plan should be verified by an analytic solution. This part of the work involves evaluating the grasp plan with consideration for the limitations of the hardware and target scene.

By combining both methods, the approach can leverage the strengths of each: using a DNN to improve sampling efficiency and an analytic method to guarantee availability, and outputting a grasp plan with available parameters. The advantages of deep learning can be applied in the field of object detection and grasp plan proposal, similar to object detection and key point detection in the visual area. The network can help simplify the environment for the analytic approach. The analytic solution can be applied in a simplified environment and used for evaluation, providing a more precise and available result for the grasping task.

6.2 Deep Learning Based Grasp Planning

Building on the concepts introduced in the previous section, this section will detail the design of the deep learning-based grasp planning approach. There section can be introduced with four subsections: 6.2.1 key parameters of grasp plan, 6.2.2 grasp plan framework, and 6.2.3 training data and training details.

6.2.1 Key Parameters of Grasp Plan

Several key parameters need to be involved in a grasp plan for a robotic end-effector, including position, and control parameters. For some common deep learning-based work, the output grasp plan is defined as shown in Figure 6.1 (a) and includes five parameters: x, y, $width$, $height$, and θ. However, for the requirements of a dexterous hand, these definitions can be further optimized.

In most common grasping cases, the end-effector is a parallel gripper that cannot change the $height$ parameter corresponding to the distance between the fingers on one side. This only works for a few dexterous hands with a layout like Robotiq three-finger hand. The $width$ parameter aims to guide the opening distance of the

end-effector. However, according to actual results, this parameter usually does not function in most cases. The usual strategy is to open with the largest range or a suitable distance according to obstacle information. The θ is usually generated with some sparse angles, and choosing too many angles may affect the efficiency. Typically, the angle step is set to a value like 10°. With these considerations, the grasp box is limited and cannot provide enough grasp plan information for a dexterous hand. A new grasp plan presentation is necessary for dexterous hands.

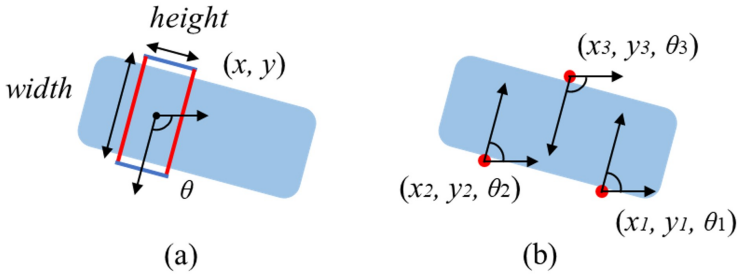


Figure 6.1: Grasp plan with grasp box and contact points

As analyzed in the analytic section, the presentation of contact points can cover the information required by a dexterous hand. Using contact points can help make the plan more flexible. Therefore, the grasp plan should include the position and direction of contact points. A grasp plan with n fingers can be presented as (x_i, y_i, θ_i), $i = 1, 2, \cdots, n$. Figure 6.1 (b) shows a grasp plan using contact points. Considering the output, the location (x_i, y_i) of each contact point is output directly by the neural network, and the direction θ_i of each contact point is determined by the normal vector of the contact surface. Then the grasp plan consists of two parts: the contact points by the neural network, and the normal vector direction of each contact point. With these considerations, the output of the framework is decided.

6.2.2 Grasp Plan Framework

As a visual-based approach that leverages the capabilities of the DNN, the object detection and segmentation component should be included within the framework. Compared to an end-to-end solution, separating the framework into several parts can provide more information and explain the work process. It can also better integrate with the analytic approach. As the key parameters required by the grasp plan, three main components are necessary: object detection and segmentation, grasp point generation, and grasp plan evaluation. The structure of the network is illustrated in Figure 6.2.

The object detection and segmentation component takes the target scene as input and outputs the detection and segmentation results for the next component.

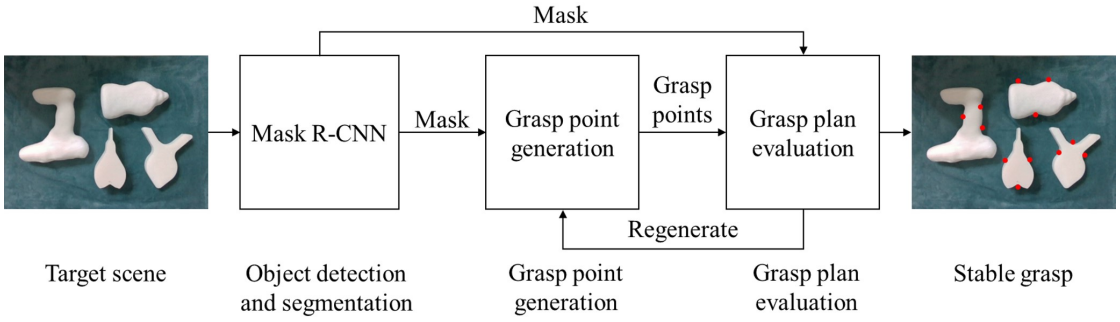


Figure 6.2: Framework of visual-based grasp planning

The resulting mask is then used as input for the next module, helping to remove the background and ensure that grasp planning focuses on the target object. The mask of the target object is also used in the grasp plan evaluation component, where it serves as the polygon for force-closure calculation. The object detection and segmentation module takes advantage of Mask R-CNN [He et al., 2017], training the network based on a pre-trained model and referring to the config file of $e2e\_mask\_rcnn\_r\_50\_fpn\_1x.yaml$ [Massa and Girshick, 2018]. The output dimension is changed with the object types in the experiment.

The grasp point generation component takes the mask as input and outputs grasp points, which are the contact points used for the grasp plan. The number of contact points is decided by the number of fingers. These points are fed into the grasp quality evaluation component. The model is built based on a Visual Geometry Group (VGG)16 model [Simonyan and Zisserman, 2014] and takes reference from facial keypoint detection work [Alex, 2021].

Compared to a typical keypoint detection network, the main difference is that for an object, there are unlimited numbers of grasp solutions, whereas facial keypoint only include one ground truth. This key difference must be described by the loss function, meaning that the loss function that calculates the distance between predicted results and label results cannot meet all requirements. The loss function should introduce the grasp quality evaluation component to check whether the grasp plan is suitable for a dexterous hand from several perspectives and score it, including four parts: 1) point distance; 2) point location; 3) grasp stability; 4) grasp dimension.

The point distance should guarantee these three points has enough distance that can be regarded as a three finger grasp. The point location means the point should be on the edge of the target object. The grasp stability can be evaluated by an analytic solution, where the point direction is calculated from the normal direction of the contact point, and stability is determined according to force-closure criteria. The grasp dimension is used to ensure that the grasp plan is suitable for a dexterous hand and should be designed with different types of end-effectors. These four parts

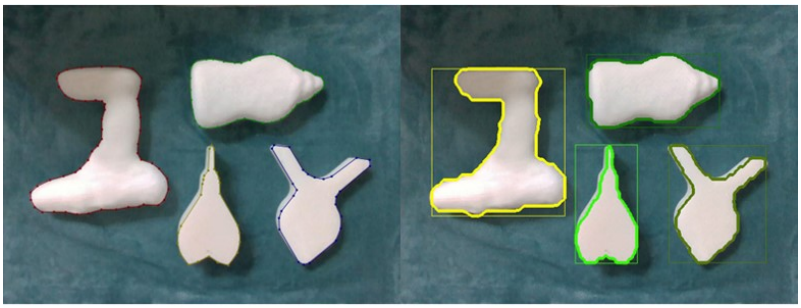
will be used in the loss function to help train the grasp point generation network. All four of these points are considered at four levels in the loss function to improve training results.

The grasp plan evaluation part evaluates grasp points and converts the available ones into a grasp plan that meets the requirements of a dexterous hand. This component also needs to consider the other parameters, including the Table-Top assumptions and others. When a grasp plan output by the network does not meet the requirements, it will translate or rotate the mask and input it into the grasp point generation module again until a plan that passes the criteria is found. With this module, it can guarantee that the grasp plan for the robot is available and can successfully grasp the target.

6.2.3 Training Data and Training Details

As introduced in the grasp plan framework, there are two networks that require training samples. There are two types of samples: the mask of the target object, and the grasp points of the target object.

Figure 6.3 (a) shows the training sample for Mask R-CNN. The base learning rate was set to 0.005 and with a weight decay of 0.0001. The number of images per batch was set to 2 and the training steps were set to 180,000 steps. The segmentation result on the validation set shows good performance on segmentation. The segmentation result is shown in Figure 6.3 (b).



(a) (b)

Figure 6.3: Training sample and test result of the object segmentation model

Figure 6.4 (a) shows the training sample for the grasp point generation model, and Figure 6.4 (b) shows the grasp points result. The base learning rate was set to 0.0001 and the training steps were set to 15,000 steps. The number of input data per batch was set to 1. The combination of distance loss and analytic loss is important in this framework. The weight of distance loss represents the importance of the sample data, while the weight of analytic loss represents the importance of analytic analysis. To obtain an optimized loss function, a comparison with these two ratios was

performed, with the ratio range changed from 0% to 100%, and finally obtaining the best result with 50% L1 loss and 50% analytic loss. The loss value for each criterion was defined through the training process. The comparison between 100% L1 loss and the combined solution can be seen in Table 6.1. The success rates are compared in both the training set and the test set and are evaluated with four analytic criteria. The function of the training sample guarantees basic performance, while analytic loss improves upon it.

(a) (b)

Figure 6.4: Training sample and test result of grasp point generation model

Table 6.1: Grasp test statics

Type	Training set				Test set			
	Point distance	Point location	Grasp stability	Grasp dimension	Point distance	Point location	Grasp stability	Grasp dimension
L1 loss	90%	32%	30%	24%	100%	81%	76%	67%
Combination	95%	33%	30%	26%	100%	86%	81%	72%

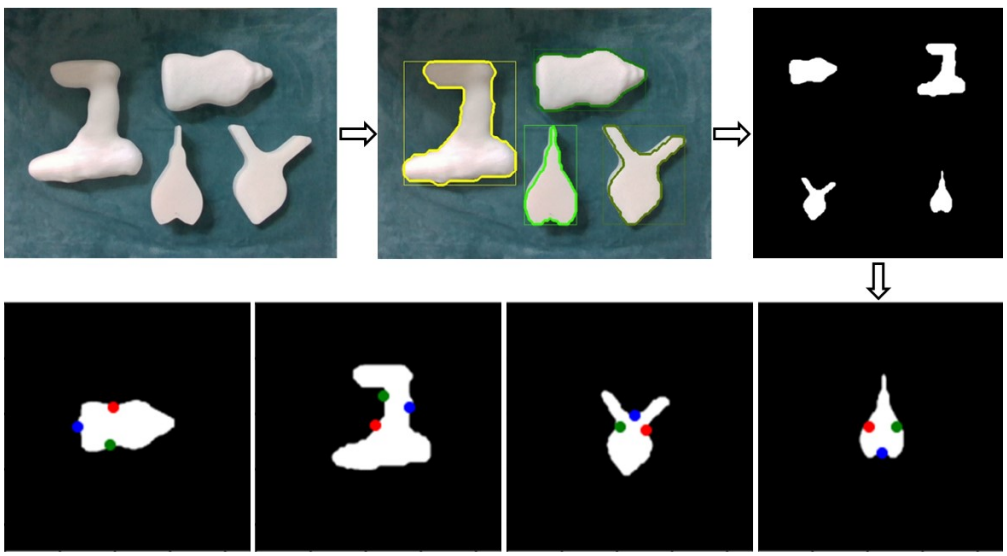


Figure 6.5: Grasp planning process

The entire grasp planning process can be seen in Figure 6.5. The steps include

inputting visual data, segmenting it with Mask R-CNN, obtaining grasp points, and finally obtaining a grasp plan. The training machine has an Intel i9-9900KF CPU and a GTX 1080 Ti GPU. The framework is built with PyTorch 1.4.0, CUDA 10.0.130, and Python 3.6.9.

6.3 Performance Evaluation

The main objective of the evaluation process is to verify the real-world capability of the grasp planner. An experiment was conducted to verify the grasp result with a real dexterous hand.

For the objects in the experiment, many different types are present in a Table-Top scene. According to the shape and dimensions of the DoraHand, twelve objects with certain characteristics were selected from the YCB object set and some common shape objects. The dimension range is 50 mm - 360 mm, and the shapes include cube, cuboid, triangle, arch, bottle, and special shapes. These objects can be seen in Figure 6.6. The model was trained with all of these objects.

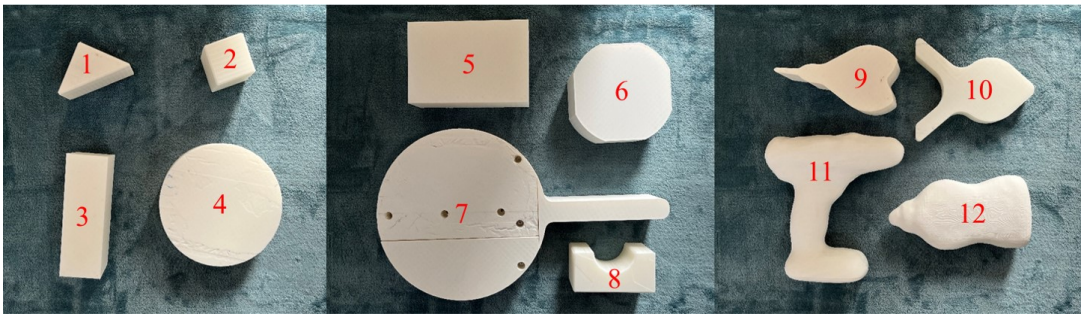


Figure 6.6: Twelve types of objects for grasp planning test

Table 6.2: Grasp test success rate

Total	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
	80%	100%	100%	100%	90%	100%
93.3%	Item 7	Item 8	Item 9	Item 10	Item 11	Item 12
	100%	90%	90%	80%	100%	90%

There grasp success rate criteria were used for this learning approach. It is the rate at which the grasp plan can be executed and successfully grasp the item. The grasp test was conducted 120 times, with 10 times for each item. These two criteria are presented in Table 6.2. The total success rate is 93.3%, and there are some failure grasp plans. Figure 6.7 shows three typical failure cases. In (a), the distance between the points is too close, and the force direction of point two has a wrong value. This is caused by the jagged edge of the mask causing the wrong normal vector direction. In

(b), point two is inside the target and has the wrong direction. This is caused by an error in the function for checking whether it is on the edge. In (c), the item cannot be grasped with a two finger grasp and the direction has errors to be executed. The force-closure can be achieved with this grasp but cannot be completed by hand. These errors show that, the object detection and segmentation, and the grasp plan evaluation part need to be improved with higher precision, and there are still missing some criteria for the grasping capability of the hand. For further development, evaluating the grasp plan in a simulation environment with kinetic parameters may help to get a more comprehensive evaluation. Some successful grasp cases can be seen in Figure 6.8.

Figure 6.7: Some failed grasp plans

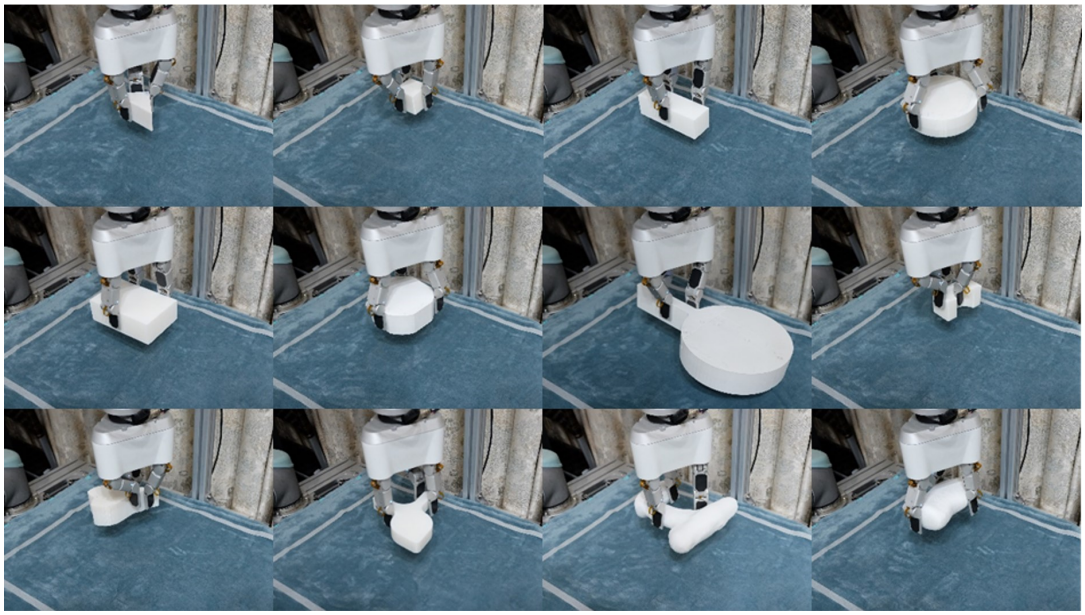


Figure 6.8: Some typical grasps with DoraHand

6.4 Conclusion

The visual-based grasp planning approach defined a grasp plan representation with contact points. The training result performed well in the experiment and demonstrated grasp plans with a good success rate. With the input of visual data, the combination of analytic and data-driven approaches can provide a solution for grasp plan cases in a Table-Top scene. With the grasp plan used in this work, a dexterous hand can be used in a grasp planning task and able to take advantage of multiple DOF. This work indicates the importance of evaluation work with more detailed parameters, and more training data can help to get better results. The following part will focus on tactile-based stable grasping, and the next chapter will mainly introduce work related to the dataset for stable grasping.

Part IV

Stable Grasping

Chapter 7

Visual-Tactile Dataset

Stable grasping is a crucial aspect of robotic manipulation, and evaluating grasp stability is essential for achieving it. To accurately evaluate grasp stability, data related to the grasping status is necessary. The features of the grasp status need to be extracted from the data and will directly influence the evaluation approach. Building upon the hardware designed in the previous chapter, this chapter focuses on constructing an open-source dataset that includes visual, tactile, and other data that can be used to evaluate grasp stability. This chapter is divided into four sections: Section 7.1 introduces the design of the dataset and related experiments. Section 7.2 presents the detailed process. Section 7.3 provides an analysis of the dataset. Section 7.4 summarizes this chapter.

7.1 Dataset and Experiment Design

Before creating a suitable dataset, it is necessary to determine its scope and design the experiment. This section is divided into two subsections: 7.1.1 dataset definition, and 7.1.2 experiment design.

7.1.1 Dataset Definition

The first step in developing a dataset for robotic manipulation is to define the data that needs to be included. This dataset is intended to evaluate grasp stability and provide comprehensive data to help analyze the entire process. Ideally, it should collect sufficient data to reconstruct the environment and recover the grasping process.

To reconstruct the target environment, a fundamental element is a 3D space. The 3D models of the end-effector, robot arm, and target object can be obtained from their design models or object set. The RGB and depth visual sensors should be used to capture the entire grasping process. These data provide the basic information for the

visual reconstruction of the grasping process.

In the visual data, the end-effector may be occluded during grasping. Therefore, joint data of each joint must be recorded to recover the grasping process. Additionally, tactile data is a crucial component, and the end-effector should provide tactile data throughout the entire process and meet the requirements for recording the tactile data between the end-effector and the object. Based on these considerations, the dataset should include the following data:

1. The basic experimental conditions.
2. RGB and depth images and videos of the entire grasping process.
3. Joint values of the joints in the dexterous hand.
4. Tactile data from the dexterous hand.

7.1.2 Experiment Design

With the requirements of a dataset for robotic manipulation and to facilitate the grasp stability evaluation, a common grasping scenario on a table is selected for the experiment. The experiment is designed with some hardware including an Eagle Shoal dexterous hand, two RealSense SR300 cameras, and a UR5 robot arm. These components are arranged around a 600 mm x 600 mm platform, as shown in Figure 7.1. A layer of sponge is placed on top of the platform to reduce the effects of reacting forces from the surface, and a soft flannel sheet covers the platform to prevent interference from light reflection. The UR robot arm is positioned at the back of the platform with the Eagle Shoal attached to it. One SR300 camera is placed on the opposite side of the UR arm to capture the front view of the grasping process, while the other SR300 camera is located to the left of the platform to record the lateral view.

The Eagle Shoal is a dexterous hand designed with a two-part finger and the palm layout 2, and is shown in Figure 3.5. It features three fully-actuated fingers and one fully-actuated palm, providing a total of eight DOF. The hand is equipped with seven tactile sensor modules, including a total of sixteen film sensor units, and their distribution is similar to that of the DoraHand. These sensors measure the normal force applied to them in mN. The hand can transmit sensor data at a rate of 40 Hz via a CAN bus. Figure 7.2 shows the positions of the film sensors and position sensors. The joint sequence is the same as that in the position data file. The red number on the fingers and palm represents the film sensor positions and numbers in the tactile data file. Table 7.1 displays the main parameters of the robot hand.

Two SR300 RGBD cameras were used to record the grasping process from two different vertical angles. However, since the cameras use the same structured light to

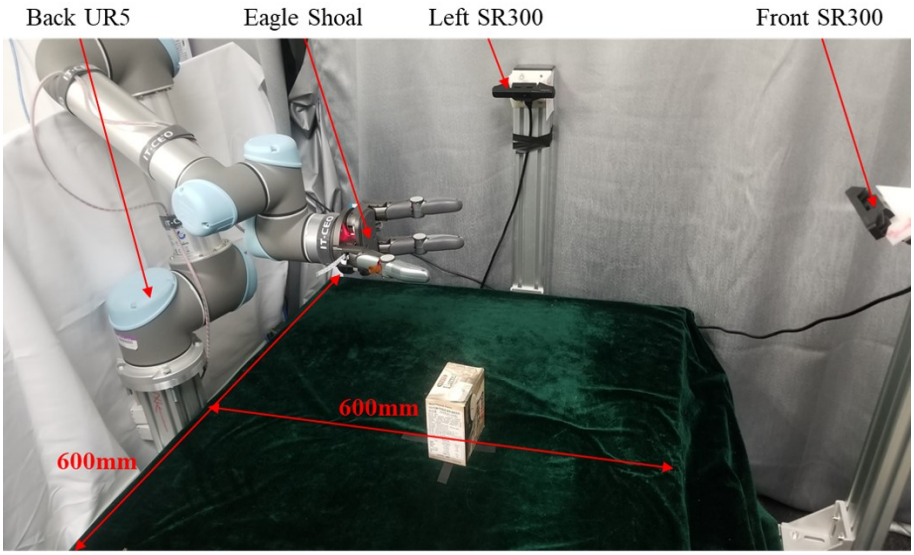


Figure 7.1: Experimental setup for data collection

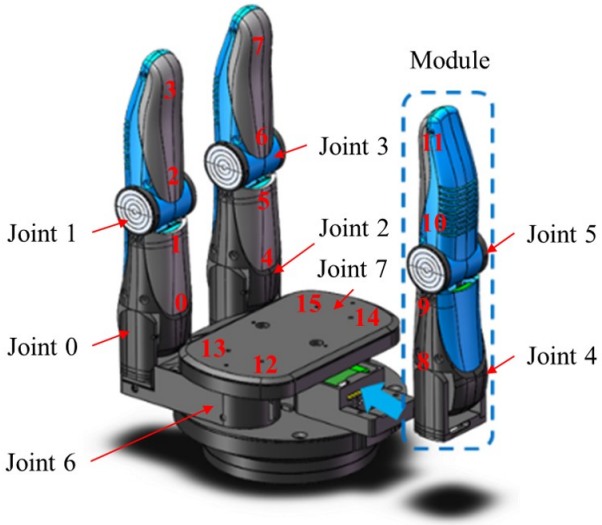


Figure 7.2: The main structure of Eagle Shoal

Table 7.1: Basic parameters of Eagle Shoal

Dimension L-W-H	Weight	Joint number & range	Joint speed & precision
128mm-100mm-185mm	790 g	8 / ± 90°	72°/s / ± 0.3°
Film sensor number & range	Input power	Communication type & rate	Encompassing grip payload
16 / 10 N	12 V / 2 A	CAN / 40 Hz	2 kg

obtain depth data, interference between the two cameras can occur. To prevent this, an opaque material was used to block the lateral camera's structured light projector during the experiment. Only the front camera's depth video and both cameras' RGB videos were recorded. The resolution and frequency of the RGB and Depth videos were set to 640 x 480 and 18 Hz, respectively.

During the experiment, point clouds were used to segment the target object and calculate its dimensions and centroid for grasping. The grasp plan in this experiment was simple: the centroid was selected as the grasp center, and applied in different grasping approaches from the side or top. The strategy used in the grasping process involved using position control to reach the pre-grasp pose and achieving grasping with current control. The recording part of the experiment was automatically triggered by the task scheduler. The main program controlled the experiment procedure, sent the control commands to the UR5 arm via socket, and sent control commands to the Eagle Shoal via CAN bus. The software environment used was Ubuntu 14.04 and ROS Indigo, which consists of three parts: visual, data recording, and task scheduling.

To meet the requirements of the experiment, objects were carefully selected based on several criteria: 1) grasp dimensions within a range of 150 mm; 2) different shapes such as cylinders, rectangles, and special shapes; 3) different surface materials; 4) ability to contain different payloads. Ten objects were selected from the YCB object set and daily life objects based on these criteria. These objects are displayed in Figure 7.3.

Figure 7.3: Selected objects for visual-tactile dataset

7.2 Experiment and Data Format

With the defined scope of the dataset, the dataset can be created using the hardware and experiment. This section includes two subsections: 7.2.1 experiment, and 7.2.2 data format.

7.2.1 Experiment

Various factors such as shape, size, weight, and grasping gestures need to be considered to construct a comprehensive dataset. For the selected set of objects, different grasping gestures were defined by grasping from three directions: back, right, and top. And grasping status is changed by adjusting the object weight, grasp force, and grasp position. With these conditions in place, the experimental process was carried out with a specific procedure, as illustrated in Figure 7.4.

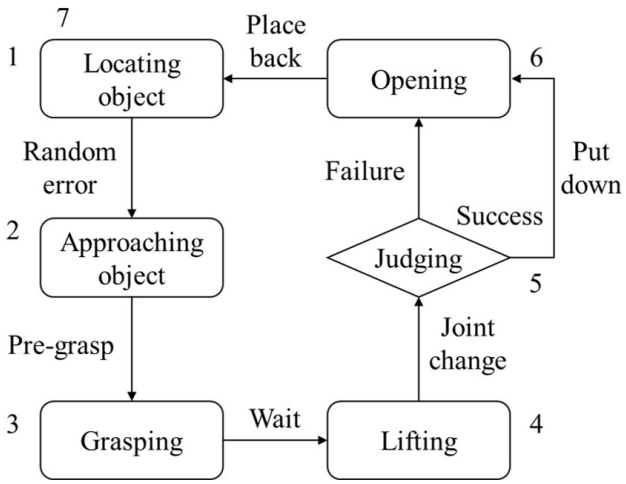


Figure 7.4: Grasping experiment pipeline

1. Place the object at the center of the table and use the front camera to capture point cloud data. Filter, segment, and locate the target object's position using the point cloud.

2. Add a random error within $\pm$ 5 mm to the object's half-height center position to generate a grasp plan and control the robot arm to approach it.

3. Based on the object's dimensions, control the robot hand to assume a pre-grasp gesture using position loop mode, then switch to current loop mode to grasp with a predefined current.

4. Wait for 1 second to ensure stability, then lift the object with the arm at a speed of 20 $\mathrm{mm/s}$.

5. After the robot arm has lifted to a certain position, judge the grasp status based on changes in each finger's position. If the difference in finger joint angle exceeds a threshold during lifting, it indicates that the object was dropped during the process and this grasp is labeled as a failure; otherwise, it is labeled as a success.

6. If the label is a failure, open the hand directly and prepare for the next grasp. If the label is a success, for light objects, place them down and open the hand; for heavy or easily slipped objects, open the hand and drop them directly.

7. Place the object back to the center of the table, control the robot arm to return to its initial position, and wait for the next loop.

During the experiment, it was observed that the soft surface of objects and the structure of the tactile sensor made it challenging for the fingers to obtain significant sensor values during grasping. To ensure noticeable value changes, grasping gestures were designed to involve the use of only three fingertips. Additionally, in cases such as grasping cylinders using a power grasp pose, it provided reliable sensor feedback, eliminating the need for tip grasping poses.

7.2.2 Data Format

After collecting data from the sensors, several processing steps were performed. The data was organized and saved into a folder path with a specific name format, such as "object type/current value_object weight/grasp direction/attempt sequence/filename". Within the folder, three types of data were stored: ".txt" files containing text data like force and label; ".jpg" files containing image data captured at different directions and time steps; ".mp4" files containing video data that recorded the entire test process. In the case of depth images in CV_16UC1 format and videos in 8-bit and 3-channel format, for the depth video, the high 8-bit data was saved in the red channel, the low 8-bit data in the blue channel, and to enhance video clarity, the middle 8-bit data was added to the green channel.

A total of nearly 3,000 test rounds were conducted, resulting in a dataset comprising 2,550 sets of valid visual and tactile data. The dataset can be accessed on GitHub [Wang, 2018] for further exploration and analysis.

7.3 Dataset Introduction

After conducting the experiment and performing basic processing, the dataset was completed and made available as open-source. This section introduces the compo-

sition of the dataset and presents some detailed data. It includes two subsections: 7.3.1 statistic of the dataset, and 7.3.2 raw data.

7.3.1 Statistic of the Dataset

The dataset consists of data with various conditions, including weights, forces, directions, and object types. It can be divided into two sets: Empty and Non-empty. The main difference between these two sets is whether the objects are empty or non-empty. In the test of empty cases, all of these grasps were successful due to the low weight of the objects. In the test of non-empty cases, include both successful and failed attempts. Each test contains four kinds of data, and the data statistics are shown in Table 7.2.

1. T1 represents the tactile sequence and label document data. The tactile data consists of 16 channels of data from the tactile sensors, recorded over 400 time steps for a duration of 24 seconds. The label document records the time steps at which different commands are sent, and the final success or failure label for each grasp attempt.
2. T2 represents the position sequence data. The position sequence data consists of 8 joint positions, recorded over 400 time steps for a duration of 24 seconds.
3. I represents the image data. The image data comprises four images for each grasp attempt, with two time steps captured from each of the two cameras.
4. V represents the video data. The video data includes two RGB videos and one Depth video for each grasp attempt.

Table 7.2: Dataset statistics

Part	Type	Weight	Current/mA	Direction	Trail	Data type	Total
Empty	10 objects	empty	50/100/150	back/right/top	10 times	T1/I	900 sets
Non-empty	10 objects	half/full	50/100/150	back/right/top	10 times	T1/T2/I/V	1650 sets

Figure 7.5 presents the number of successful and failed grasps for each object in the dataset. The overall success rate for the 2,550 grasp attempts is 66.27%. Based on the dataset statistics, objects with a cylindrical shape, such as Cola, Crisp, and Tomato, are relatively easier to grasp. Additionally, surface deformation plays a significant role in the grasping process, as objects with less deformation, such as Cheez and Sugar, exhibit a higher success rate. For some objects like Cheez and Water, the number of test attempts was relatively small, and part of the test was not performed to protect the hand, because the weight of the object in the half or full state was too heavy to grasp in some directions.

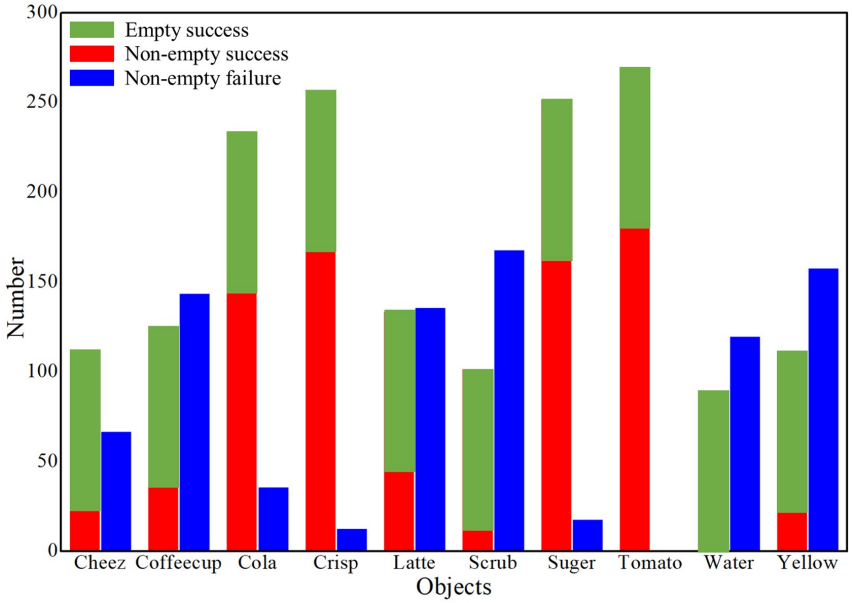


Figure 7.5: Success and failure data in the dataset

7.3.2 Raw Data

This subsection provides an introduction to the dataset by showcasing examples of different types of raw data. Since different gestures have a notable impact on tactile data, there is a comparison to illustrate and explain the variations in the data. Figure 7.6 and Figure 7.7 display two sets of tactile data along with corresponding time labels. In both figures, the horizontal axis represents time in s, while the vertical axis represents the tactile value in mN. The acquisition rate for tactile data is 16.7 Hz.

Figure 7.6 shows one set of data belonging to the Empty set. The path of the data is Cola/150_26/right/8/tactile.txt, which provides specific information about the object type (Cola), current value (150 mA), object weight (26 g), grasp direction (right), attempt sequence (8th), and data type (tactile sequence data). Three time steps are labeled with the corresponding label document. These three vertical lines in the figure represent the time pf sending the position loop, current loop, and lifting commands respectively. Obvious changes in the tactile values can be observed around each labeled step. After several steps of the green position loop line, the hand gradually makes contact with the object, resulting in an increase in tactile values. The object may shift position, leading to corresponding changes in the tactile values, as indicated by sensor 10. Following the yellow current loop line, the hand switches to current loop mode, resulting in a further increase in tactile values before reaching a stable status. Once the red lifting line is reached, tactile values begin to change again, with some sensors such as sensor 9 exhibiting rapid changes. After several

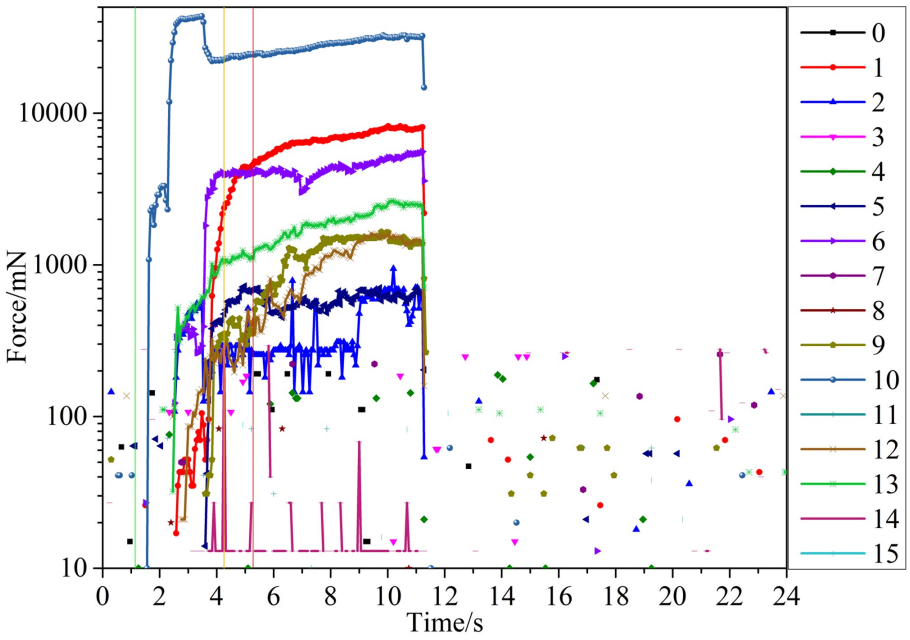


Figure 7.6: Visualization of raw tactile data Cola/150_26/right/8/tactile.txt

seconds, the arm moves down and the hand opens, causing a sudden drop in tactile values. If part of the object touches the plane earlier during the object moving down, the tactile data increase when there is a touch and drop to zero when fully opened, like the changes in sensor 2. For some heavy objects with unstable state, the hand is set to open directly at the highest position, and the tactile data will drop earlier.

Figure 7.7 shows one set of data belonging to the Non-empty set. The path of the data is Scrub/150_866/top/5/tactile.txt, which provides specific information about the object type (Scrub), grasping current value (150 mA), object weight (866 g), grasp direction (top), attempt sequence (5th), and data type (tactile sequence data). It is a failed attempt where an object was dropped during the lifting process. Unlike the label document in the empty set, the red line in Figure 7.7 represents the time when the arm reached the highest position. This label can help to find the time range of grasping faster. A significant change in tactile values can be observed in sensor 3 at around 7 s, between the yellow and red lines. This change indicates the point at which the object began to drop. The tactile values recorded after the red line are a result of the current loop grasping with no object present, causing the sensors to make contact with the hand itself.

The tactile data in these two figures clearly highlights the significant changes in tactile data at each key time step, and show some differences between the Empty and Non-empty data. These changes and difference serve as crucial indicators for understanding and analyzing the grasping process.

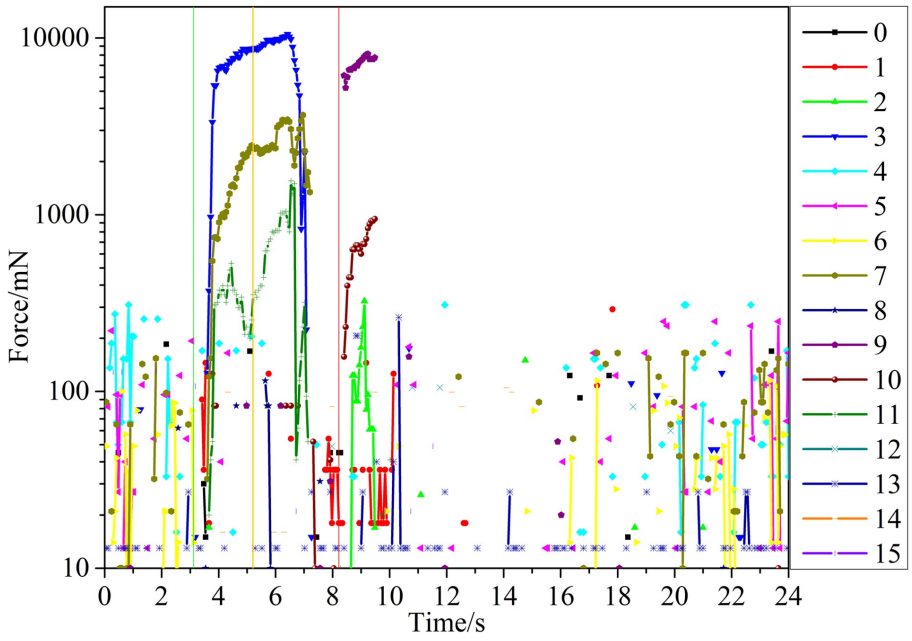


Figure 7.7: Visualization of raw tactile data Scrub/150_866/top/5/tactile.txt

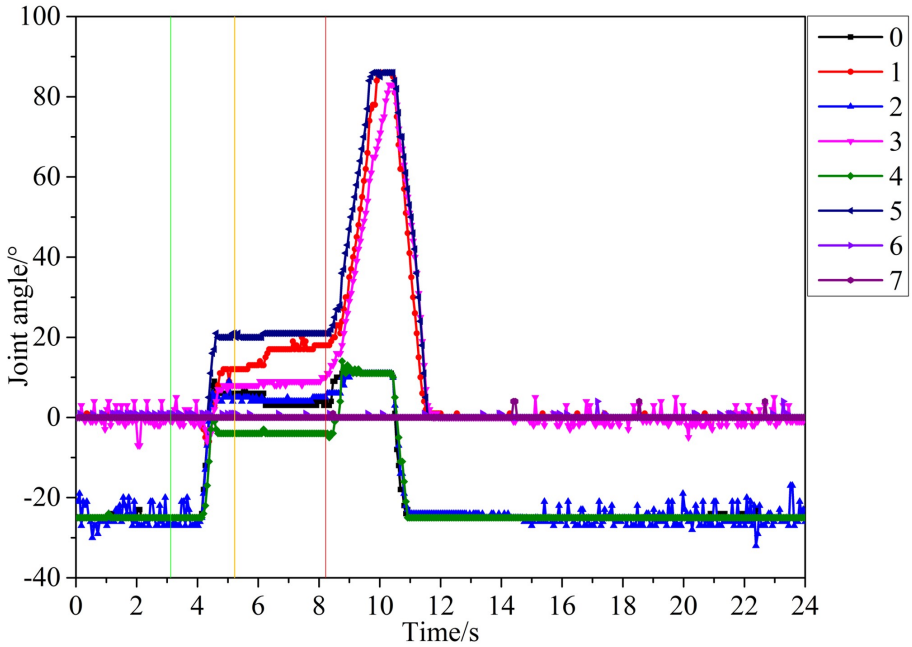


Figure 7.8: Visualization of position data Scrub/150_866/top/5/pos.txt

In addition to tactile data, analyzing joint positions in the robot hand provides valuable insights into the grasp status. Figure 7.8 illustrates 8 channels of joint position values over a duration of 24 seconds. The data shown in the figure, Scrub/150_866/top/5/pos.txt, corresponds to the same attempt as Figure 7.7, where "pos" represents the hand joint position data. Upon observing the position data, it can be noted that after the position loop line, each joint angle changes rapidly and stabilizes after several steps. With the transition to the current loop mode, joints 1 and 3 exhibit small position changes before achieving a new force balance. During the interval between the current loop and the highest position, around 7 s, the object drops and the hand gradually closes with the current loop, corresponding to the force change observed in Figure 7.7. By analyzing the joint position data, one can easily identify the status of the robot hand at each time step, which can greatly assist in determining the grasp status.

The image data contains four images from two different directions at two different time steps: two at the start time of the test, and the other two at the time when the current loop command was sent. These images display the relative positional relationship between the hand and the object, with a sample image shown in Figure 7.9.

Figure 7.9: Image captured by two cameras Scrub/150_866/top/5/*.jpg

Furthermore, the video includes two RGB videos and one Depth video. These videos can be utilized to reconstruct the scene with a model of the dexterous hand,

robot arm, and objects. Part of the video data is shown in Figure 7.10, where rows one to three are video screenshots of left, front, and front depth videos, respectively.

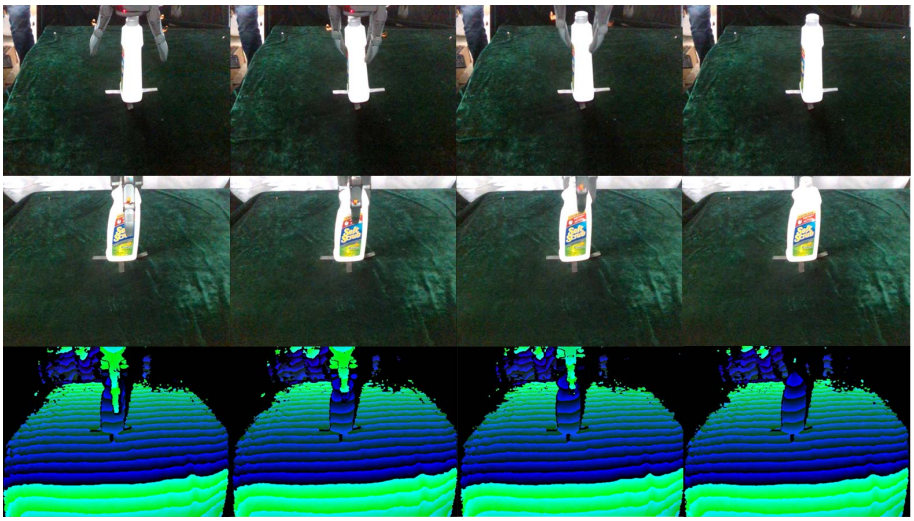


Figure 7.10: Several frames in video Scrub/150_866/top/5/*.mp4

The combination of tactile, label, position, image, and video data offers a comprehensive understanding of the grasping process and contributes to a more accurate evaluation of the overall grasp performance. With the various types of data, there are many potential applications. For instance, tactile data can classify objects, visual data instruct the robot on grasping techniques and maintaining stable grasps, the combination of tactile and position data can analyze micro movements during grasp, and the combination of tactile and visual data can evaluate grasp stability. This dataset enables researchers to leverage the strengths of different data types to achieve tasks related to robotic manipulation.

7.4 Conclusion

This chapter presents the construction and introduction of the visual-tactile dataset created using the Eagle Shoal dexterous hand. With a detailed explanation of the creation process and composition of the dataset, users can gain a comprehensive understanding of its contents. This dataset includes 2,550 sets of data that may be useful for stable grasping and tactile sensing applications. Based on the statistics, the dataset shows some key factors during the grasping process and demonstrates its potential usage for stable grasping. While it is not a perfect dataset that provides all information during the grasping process, it is a good trial that combines the visual and tactile data and focuses on the grasping process. The next chapter focuses on the stable grasping based on this dataset.

Chapter 8

Tactile-Based Stable Grasping

As robotics technology advances and expands into new applications, the ability to grasp objects has become increasingly important. This is particularly true in emerging fields such as logistics, where robots need to handle a wide variety of items. To ensure stable grasping of various objects, robots must be equipped with grasp stability evaluation capabilities, allowing them to assess and adjust their grasp as needed to ensure it is appropriate for the object.

This chapter will introduce the stable grasping function based on tactile data, using the dataset presented in the previous chapter. The chapter is divided into five sections. Section 8.1 defines the problem to be solved. Section 8.2 introduces the approach for achieving stable grasping. Section 8.3 introduces the training details and defines the experiments. Section 8.4 demonstrates the capabilities of the stable grasping function. Section 8.5 provides a summary of this chapter.

8.1 Problem Description

Evaluating the grasp status and responding to it is critical for stable grasping, and the problem is to find the evaluation criteria. Unstable grasps are caused by unbalanced force or torque, resulting in vibration or slip. Slip is an important feature of unstable grasping and can be indicated by a slip signal when relative motion occurs parallel to the contact surface. Referring to the work on slip detection can help evaluate grasp stability. By analyzing research on unstable grasping and slip, grasp stability can be evaluated on three levels [Chen et al., 2018].

1. Vibration.
2. Vibration and incipient slip.
3. Vibration and gross slip.

Vibration is the initial signal that occurs in all three phases and can be detected by a tactile sensor. To find a faster approach to detect unstable state, it is necessary to confirm the difference that happened between the force and deformation. This relationship is shown in Equation 8.1, which describes the relationship between the deformation of a thin, homogeneous plate and the corresponding forces [Stachowsky et al., 2016]. In the equation, $u(x, y)$ represents the deformation of the central plane of the plate, $f(x, y)$ represents the distribution of transverse load along the plate, and Q is the parameter related to Young's modulus, Poisson ratio, and plate dimensions. It is clear that force is highly related to the deformation of the surface $u(x, y)$. Since the change of force is related to the second and fourth derivatives of the deformation, the change in force $f(x, y)$ value is faster than other values.

$$\frac{\partial^4 u(x,y)}{\partial x^4} + 2\frac{\partial^2 u(x,y)}{\partial x^2}\frac{\partial^2 u(x,y)}{\partial y^2} + \frac{\partial^4 u(x,y)}{\partial y^4} = \frac{-f(x,y)}{Q} \tag{8.1}$$

As shown in the equation, if tactile sensors can directly measure force changes, they can respond faster than other sensors when there is vibration. They can immediately detect unstable state based on changed values [Van Wyk and Falco, 2018]. Although some visual-based sensors can assess grasp stability through deformation changes [Dong et al., 2017], these changes can only be detected after a significant change in force or displacement occurs.

Based on the tactile sensor data in the built visual-tactile dataset, an algorithm will be developed to evaluate robotic grasp stability and achieve stable grasping using these data. Since the tactile data is time series data, and the features for an unstable state are similar across different sensors, it is possible to develop a model that can generalize to other types of time series data. To detect unstable state earlier and maintain stable grasping, a grasp stability prediction function should be developed. Tactile-based stable grasping can be achieved with loop control of feedback from the grasp stability prediction component.

8.2 Stable Grasping Function

For a stable grasping function that requires grasp stability prediction and generalization, a prediction method based on deep learning is developed, and the function can be achieved with loop control. This section is divided into three subsections: 8.2.1 grasp stability features, 8.2.2 grasp stability prediction framework, and 8.2.3 stable grasping algorithm.

8.2.1 Grasp Stability Features

To evaluate grasp stability, it is important to identify features of a grasp, including both stable and unstable grasp. Since force guarantees stable grasping, changes in force value are a primary feature. However, when slippage occurs with low frequency and sampling is also at low frequency, the force-closure check may still be achieved at all sampling times. Therefore, changes in force values over a certain period are necessary for evaluating grasp stability.

Following the idea of evaluating stability based on force changes over a certain period, features in the frequency domain can also be helpful. The Fourier transform of force data is shown in Equation 8.2 [Van Wyk and Falco, 2018]. While changes in the joint force $f(t)$ may be similar, component forces $F(w)$ at different frequencies can help to show significant differences when the grasp becomes unstable.

$$F(w) = \int_{-\infty}^{\infty} f(t)e^{-jwt}\,\mathrm{d}t \tag{8.2}$$

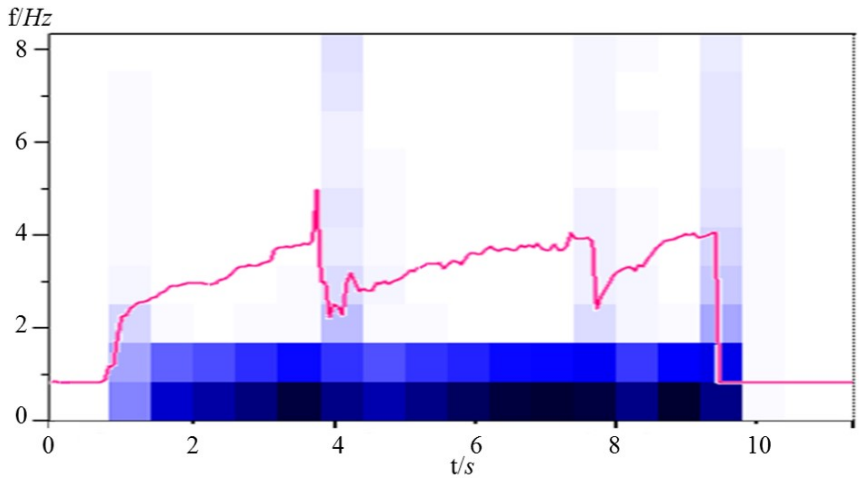


Figure 8.1: Force data Short-Time Fourier Transform (STFT) result

Figure 8.1 shows a set of force data and the corresponding STFT results. The red line in Figure 8.1 shows the change in force, while the block in the background represents the corresponding frequency composition. These significant changes in time and frequency can be detected and classified. From the frequency components in Figure 8.1, it can be concluded that the grasp status can be distinguished using frequencies of 3 - 5Hz. Higher frequency resolution and a larger range can be even more helpful.

Through these analyses, features of grasp stability can be detected in both time and frequency domains. A grasp stability prediction solution can predict and integrate these two features for reliable and fast results.

8.2.2 Grasp Stability Prediction Framework

To maintain stable grasping, it is better to predict the status of the grasp earlier. The work in this subsection is defined as a grasp stability prediction framework. To account for changes in force values over a certain period, a model is needed that can incorporate the influence of certain steps.

LSTM is a special type of Recurrent Neural Network (RNN) that can learn long-term dependencies [Hochreiter and Schmidhuber, 1997, Sak et al., 2014]. The basic module structure is shown in Figure 8.2 [Christopher, 2015]. This type of network has been applied in speech recognition, translation, and other cases that require processing time series data [Graves et al., 2013, Sutskever et al., 2014]. Researchers have also applied LSTM for slip detection, such as Convolutional LSTM (ConvLSTM), which performs well in highly dense sensor arrangements [S. et al., 2019], another study connected the output of a Convolutional Neural Network (CNN) with LSTM to process visual-tactile data and demonstrated good slip detection results [Li et al., 2018]. Applications with new features can be explored to enhance the grasp stability prediction performance based on this strong foundation of LSTM methods.

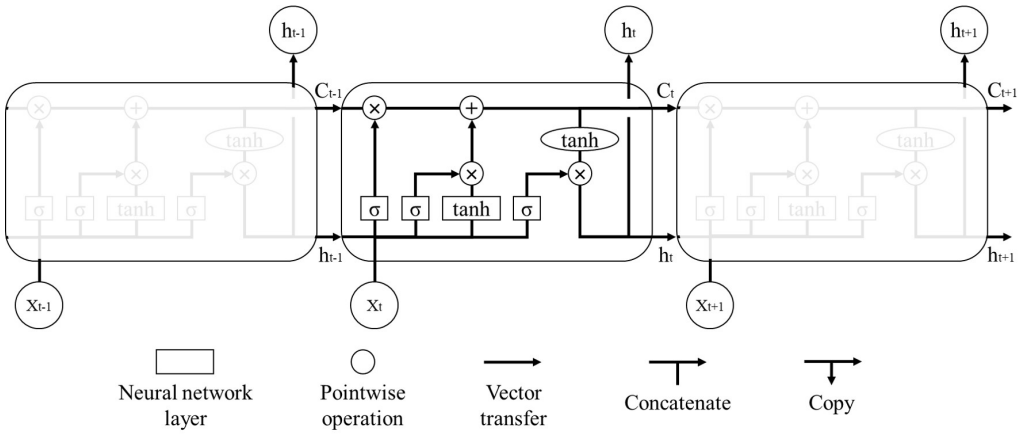


Figure 8.2: Basic unit of LSTM

The input of a basic unit of LSTM is shown in Figure 8.2, includes the data X_t, C_{t-1}, and h_{t-1}. X_t is the data input at time step t, C_{t-1} is the cell state at time step $t-1$, and h_{t-1} is the output of the cell at time step $t-1$. Each cell combines the current input and previous output, which realizes the capability to learn long-term dependencies. The grasp stability prediction network can take the tactile sensor data as the input and output the grasp stability prediction result.

Due to the features in the frequency domain, using STFT to extract features and input them into the network is worth trying. With a suitable candidate model, a duration of force and STFT data are input into an LSTM to extract features and output prediction results. It is expected that the combination of STFT and LSTM will

produce better results. The structure of the LSTM model is outlined in Figure 8.3, with inputs from x_{t1} to x_{tn} representing several frames. After comparing different output methods, the last step output is taken as the prediction result.

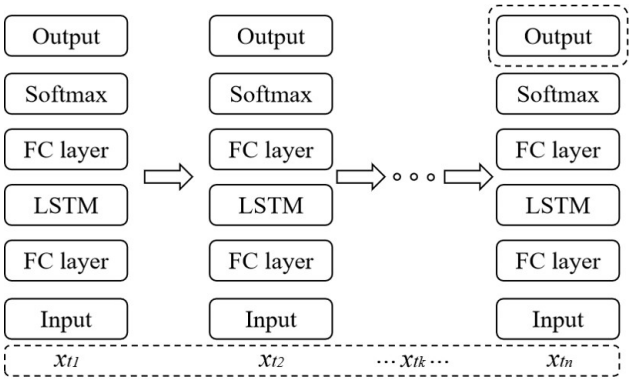


Figure 8.3: Framework of LSTM model

To combine STFT and LSTM, a variety of experiments have been performed. Figure 8.4 presents the designed framework for grasp stability prediction. The input is force data, followed by a feature extractor that includes STFT and LSTM components. These are combined in a subsequent Fully Connected (FC) layer, and the output is used as the prediction result.

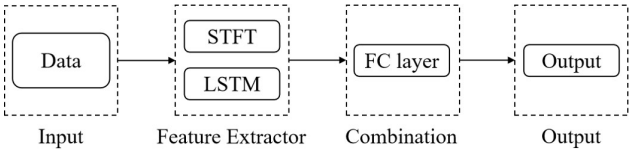


Figure 8.4: The basic structure of the network

8.2.3 Stable Grasping Algorithm

As the stable grasping function aims to ensure the stable status of the grasp, it is necessary to monitor the status during the entire grasping process. Based on the grasp stability prediction framework, the stable grasping function can be achieved with a loop control with the output status. Due to the presence of several DOF in an end-effector and the fact that grasp status is highly dependent on contact status, there are some key factors that need to be considered in this algorithm: 1) Grasp stability status; 2) Grasp plan; 3) Control type. Algorithm 5 shows the basic solution for stable grasping, and the specific loop control should be combined with the specific case.

Algorithm 5 Stable grasping

```
function STABLE GRASPING FUNCTION(forces)
    while Grasping do
        Status = Grasp stability prediction(forces)
        if Status == unstable then
            for finger in grasp_plan do                ▷ Update the control command
                Check contact points and control type
                Change values in control_command
            end for
            Send control_command
        end if
    end while
end function
```

8.3 Training Details and Experiment Setup

This section aims to refine and train the model according to the framework and design experiments for verification on the DoraHand and other time series data. To achieve these goals, this section is organized into five subsections: 8.3.1 common configuration, 8.3.2 experiment setup for LSTM, 8.3.3 experiment setup for combined models, 8.3.4 experiment setup for stable grasping, and 8.3.5 experiment setup for generalizaion ability.

8.3.1 Common Configuration

With analysis and testing on the data, common configurations can be set, including the input data and some parameters for LSTM and STFT.

The label in the datasets only represents the grasping result, so the unstable state must be manually labeled. Especially for the purpose of prediction, the label should be given earlier. Based on the procedure of the robotic hand grasping, lifting, and releasing the item, it is clear that the item will drop when the force is at 0. Since an unstable state must occur before a drop, the unstable state is labeled with the assumption that the grasp will become unstable during 20 time steps and 1.2 seconds before the drop. If the probabilities of the unstable and stable states are the same in the 20 time steps, the actual success rate of labeling should be lower than 93.75%, which is calculated with 160 steps in total.

The LSTM model was built using Tensorflow [Abadi et al., 2016]. After some initial tests, a basic LSTM core with 128 units was chosen. Grasp stability prediction was treated as a two-class classification problem and an output with softmax was produced. The model was initialized with zeros and cross-entropy was used as the loss function. Hyper-parameters were optimized based on a comparison of output re-

sults. The learning rate, which is important in the training process, was set to 0.0006 using the Adam optimizer during training [Kingma and Ba, 2014].

For the STFT, the input force data frequency is 16.7 Hz, and the target slipping frequency is 3-5 Hz. The STFT is set with a 20 time steps window size and uses a rectangular window. The window is shifted with every time step, resulting in a data size of n steps with 10 frequency bands. Because the time from the grasping phase to the opening phase is shorter than 9 seconds, a batch size of 160 is chosen to guarantee enough effective data.

8.3.2 Experiment Setup for LSTM

Grasp stability prediction is considered a classification problem. The performance of the LSTM classifier is compared with other traditional classifiers, such as Naive Bayes (NB) [John and Langley, 2013], K-Nearest Neighbors (KNN) [Altman, 1992], and Support Vector Machine (SVM) [Hearst et al., 1998], using the sixteen channels of data in the dataset as input, and set the percentages of the train set and the test set to 80% and 20%, respectively. The KNN classifier was set with $k = 3$, and the SVM kernel was chosen with Radial Basis Function (RBF).

Compared to the F_{score} used in some classification problems [Veiga et al., 2015], the prediction result of the entire grasping process and whether the output has the function of prediction are more important. Two criteria are set for evaluating the solutions: 1. the success rate, which is the rate at which the prediction matches the label; 2. the ahead drop rate, which is the rate at which the prediction is ahead of the step at which the item is dropped.

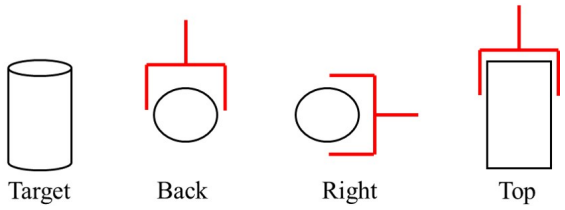


Figure 8.5: Definitions of grasp from different directions

The generalization ability of the model is important. By including data from ten objects with three different grasp directions in the dataset, the model's ability to handle differences in grasp directions and object types can be validated and optimized. As shown in Figure 8.5, different directions and items will affect the position of contact points during grasping. For example, if the model is trained with data from the back direction, it is expected to perform poorly in the top direction. Similarly, different object types also have an influence on the model's performance. The comparison experiment will sequentially remove data of different types and directions, and com-

pare the results to understand the generalization ability of the model and the impact of different data on model training.

8.3.3 Experiment Setup for Combined Models

The combination of LSTM and STFT is intended to improve the model's ability to extract features. A suitable network structure must be explored and found through different combination methods. A variety of designed models are shown in Figure 8.6. Option (a) is the original one, where force data is input into LSTM to obtain output results. Option (b) uses the frequency data as input. Option (c) uses the ten channels of frequency data and one channel of force data as input. Option (d) builds two LSTM parts to separately input the frequency data and the force data, then combines the output. The performance comparison of these four solutions can help to find the final model.

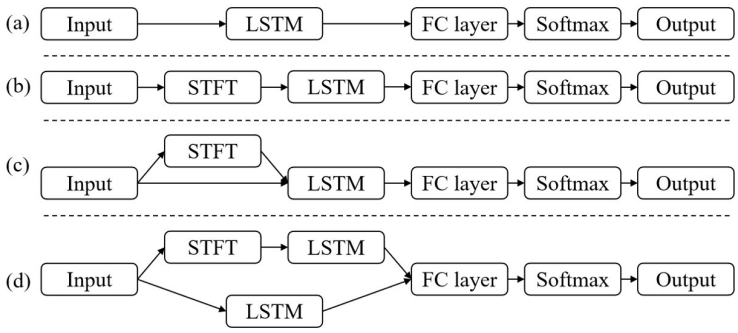


Figure 8.6: Four types of model

8.3.4 Experiment Setup for Stable Grasping

With these models listed in the subsection 8.3.3, the stable grasping function can be realized with actual dexterous hand hardware. The DoraHand, shown in Figure 8.7 (a), is chosen as the hardware. The force sensor in each finger is similar to the Eagle Shoal's sensor in the dataset, and the data frequency is set at 40 Hz. Before the stable grasping function test, the performance of different models is compared based on the sensor data acquired from DoraHand's picking process.

With the verified grasp stability prediction function, tactile-based stable grasping can be tested in real situations using the DoraHand. Four different methods representing four different grasping situations in daily life were designed: a) grasping an item with an unknown weight; b) grasping an item while taking a sudden impact; c) grasping an item while adding weight; d) rotate the item inside hand.

Referring to the stable grasping Algorithm 5, a control loop for the test is shown in Figure 8.7 (b). After the grasp stability is predicted, the current of the pj and mj

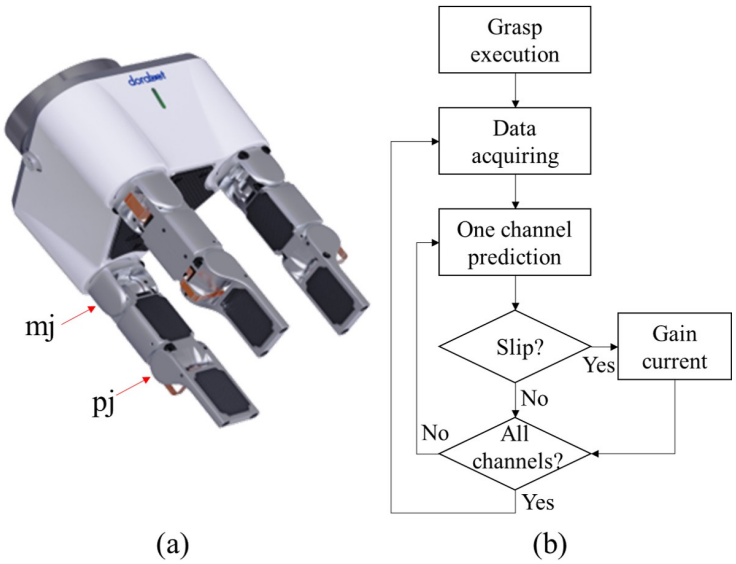


Figure 8.7: Test procedure with DoraHand

joints will be increased by 5 mA and 10 mA, respectively. The calculation time of prediction is around 4 ms for each sensor, so with 16 sensors in hand, the total time for prediction is around 64 ms. The initial current for mj and pj are 25 mA and 50 mA. The joint current for a 2 kg payload grasping is around 100 mA and 200 mA, which means unstable state should be detected for over 15 times during the process. With these tests, the tactile-based stable grasping can be verified and applied in similar applications.

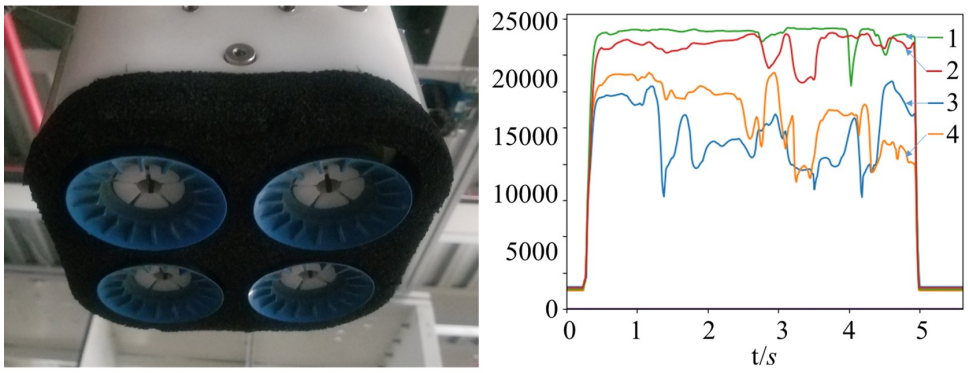
8.3.5 Experiment Setup for Generalization Ability

To test the model's generalization ability on other types of time series data, a different end-effector with a different sensor can be selected. In the industrial area, suction cup grippers are widely used and stable grasping is an important feature for protecting target items. A suction cup gripper equipped with four suction cups and four pressure sensors was chosen, as shown in Figure 8.8 (a).

Pressure sensor data was collected at a rate of 71 Hz when the robot performed a pick & place task. If the gripper cannot provide enough suction force to hold the item stable, deformation and displacement between the gripper and the item can cause changes in air pressure, especially if there is an air leak. The pressure data acquired by the sensor can show such changes and be used for grasp stability prediction.

Figure 8.8 (b) shows the change in air pressure during one pick & place task. The four channels represent four independent pressure sensors, each with a non-pressure value of around 6,400. The data is 16-bit digital data in a range of 0-65,536, with initial values are 6,458, 6,263, 6,357, and 6,458. Because the pressure unit does not

influence the result, the initial values are subtracted from the pressure sensor data and fed into the network without calibrating the relationship between the sensor data and air pressure.



(a) (b)

Figure 8.8: Suction cup gripper and pressure sensor data

8.4 Performance and Discussion

According to the experiment designed in the section 8.3, the experiment results are analyzed and discussed in this section. This includes four subsections: 8.4.1 performance of LSTM, 8.4.2 comparison of combined models, 8.4.3 verification on stable grasping, and 8.4.4 generalization ability on pressure data.

8.4.1 Performance of LSTM

The comparison result with different classifiers is shown in Table 8.1. According to the table, both LSTM and SVM perform well in terms of success rate. However, upon detailed examination, it was found that many results of SVM label on the falling edge, which means the SVM model achieves a good classification result by learning the falling edge feature. However, the falling edge means the object has already dropped and cannot help to realize a stable grasp. The LSTM approach has a high success rate of over 84% and demonstrates the best performance in the ahead drop rate with 85.88%. This suggests that these results can be used for prediction rather than detection, the LSTM has a better performance in predicting grasp stability.

One LSTM output of the data Latte/150_325/right/0/tactile.txt is shown in Figure 8.9. The blue bold line represents the predicted result, while the red bold line represents the label data. The yellow vertical line represents the time when the current loop command was sent, the red vertical line represents the time when the arm reached the highest position, and the black vertical line represents the time when the

Table 8.1: Comparison results with different classifiers

Model	Success rate	Ahead drop
NB	68.81%	65.69%
KNN	79.7%	81.76%
SVM	84.67%	66.67%
LSTM	84.60%	85.88%

object dropped from the hand. The blue bold line drops at three periods, including the vibration between 3.3 s - 4.3 s, one time between 7.6 s - 8.5 s, and one time at 9.0 s close to the hand open time. Compared with the video data, these three unstable states are easily observed: a slip when the object was lifted to the highest position, a collision with the table when the object was put down, and a drop when the hand opened. The prediction results show that the LSTM model has good performance.

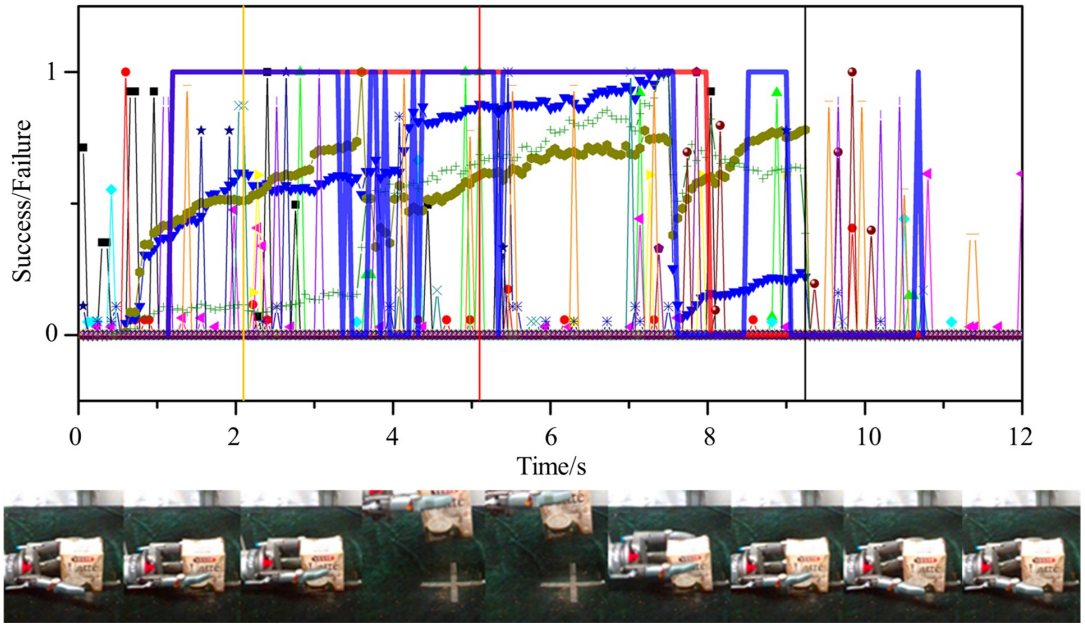


Figure 8.9: LSTM prediction result and video frames of data Latte/150_325/right/0

The comparison of directions and objects was tested separately. According to the comparison of directions in Table 8.2, the model is trained by 80% of the data in the training set direction, and the remaining 20% of the data is verified and marked as self in the table, and the other results are the verification results of the test set direction data. There was an obvious decrease when tested on different direction sets, especially for the top set, where the success rate decreased from 83.90% to 66.94% when tested on the right set. In the top set, the robot hand only grasps the object with three fingertips with sensor channels of 3, 7, and 11. In cases where the hand grasps the object with more fingers, the model only learns to predict based on three

tip channels and maybe failed to predict stability based on other channels' input.

Table 8.2: Direction comparison success rate statistics

Direction		Train set						
		Back	Right	Top	Back+Right	Right+Top	Top+Back	All
Self		87.38%	82.94%	83.90%	84.29%	82.18%	84.08%	84.60%
Test set	Back	/	78.56%	71.42%	/	79.93%	/	/
	Right	77.36%	/	66.94%	/	/	77.98%	/
	Top	76.91%	76.15%	/	77.39%	/	/	/

One of the worst cases is shown in Figure 8.10. It is a model trained with the data in the top direction, and tested with the data in the right direction, and channels 3, 7, and 11 have no obvious value during the test. Although there are data in other channels, the output prediction result is to keep the value of 0.

The result suggests that the channel difference plays an important role. To decrease its influence for a more generalized solution, the input is set with a single channel. Therefore, each set of data input into the network will include only one channel.

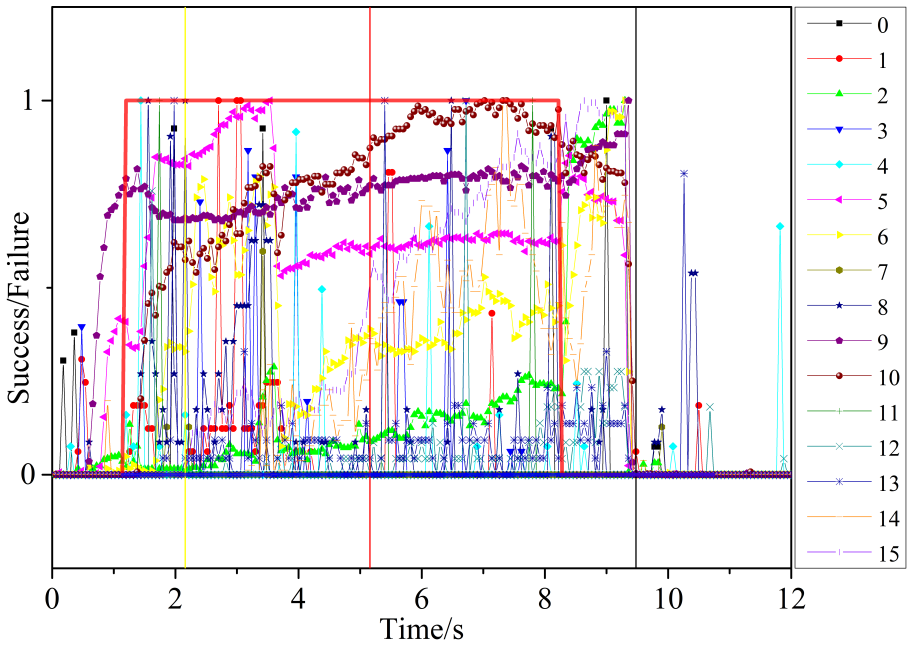


Figure 8.10: Generalization ability test in different directions with data Cola/100_657/right/9/tactile.txt

To test the generalization of the model to different objects, the target object set was removed and renamed as the self set. 80% of the self set was used for training, while the remaining 20% self set and the target object were used for testing separately. Table 8.3 demonstrates the model's good generalization ability. Some objects,

such as Scrub and Water, showed only a slight decrease in the success rate, while Coffeecup showed a slight gain. This is likely due to the presence of similar objects in the training set, which is more likely to be generalized. Figure 8.11 presents a test where an object was dropped during the lifting process, with the black vertical line ahead of the red vertical line. The prediction result shows that the prediction result changed to 0 several steps before the object dropped, which can be regarded as a good prediction result.

Table 8.3: Results of different objects

Name	Cheez	Coffeecup	Cola	Crisp	Latte	Scrub	Suger	Tomato	Water	Yellow
Self	82.89%	83.24%	82.72%	83.24%	83.47%	83.47%	83.41%	82.76%	82.86%	83.03%
Target	75.55%	83.67%	76.38%	77.89%	71.81%	81.52%	75.24%	78.65%	80.27%	79.84%

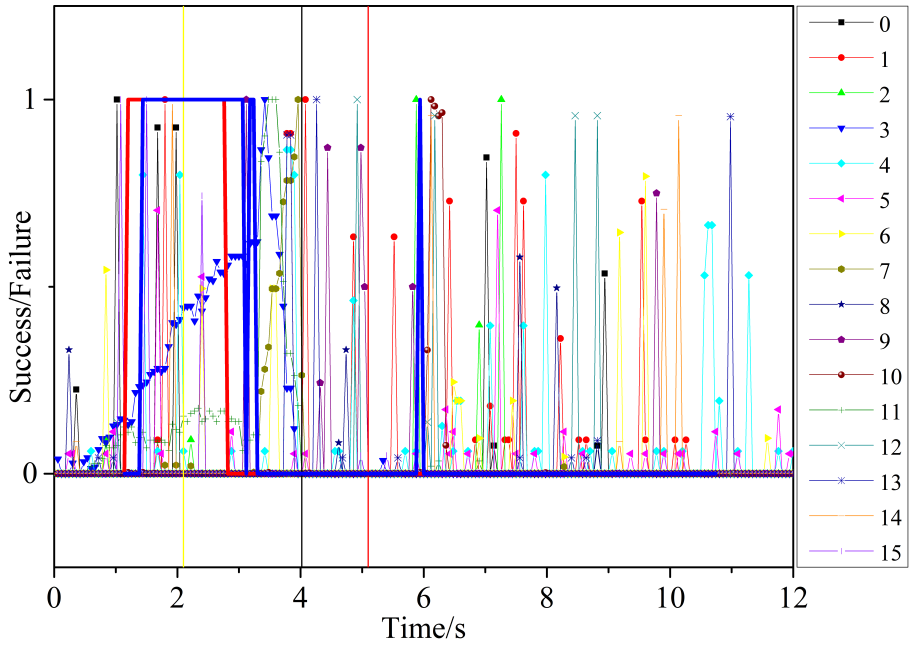


Figure 8.11: Generalization ability test in different objects with data Latte/50_620/right/3/tactile.txt

Based on the comparison of different directions and objects, it is easy that the deciding factor in the model's generalization ability is not the value, but rather the tactile channel influenced by the different contact points. This means that the positions of finger contact with the object have a significant influence. The generalization ability can be improved with a more complex dataset that includes enough grasp gestures and by training a model that excludes position differences, which can be used in all cases with tactile input.

8.4.2 Comparison of Combined Models

As shown in the test results from the previous subsection, the input of the combined model is set as single channel data. The performance comparison of the four models shown in Figure 8.6 is listed in Table 8.4. The (Data + STFT) & LSTM model has the highest success rate, while the LSTM model has the best ahead drop rate.

Table 8.4: Comparison of four models

Model	Success rate	Ahead drop
LSTM	75.27%	84.63%
STFT & LSTM	78.39%	67.88%
(Data + STFT) & LSTM	80.09%	67.50%
LSTM + STFT & LSTM	78.93%	69.25%

Predictions from the (Data + STFT) & LSTM model are shown in Figure 8.12 (a) and (b). The vertical axis represents the Boolean output of stability, where 1 indicates stable and 0 indicates unstable. The horizontal axis represents the time step, with a data frequency of 16.7 Hz. The green dotted vertical line represents the time when the current loop command was sent, the yellow dotted vertical line represents the time when the lifting command was sent, and the red dash-dot vertical line represents the time when the object dropped from the hand. The blue thin line shows the normalized force value, the red dashed bold line illustrates the labeled data, and the blue bold line represents the prediction result.

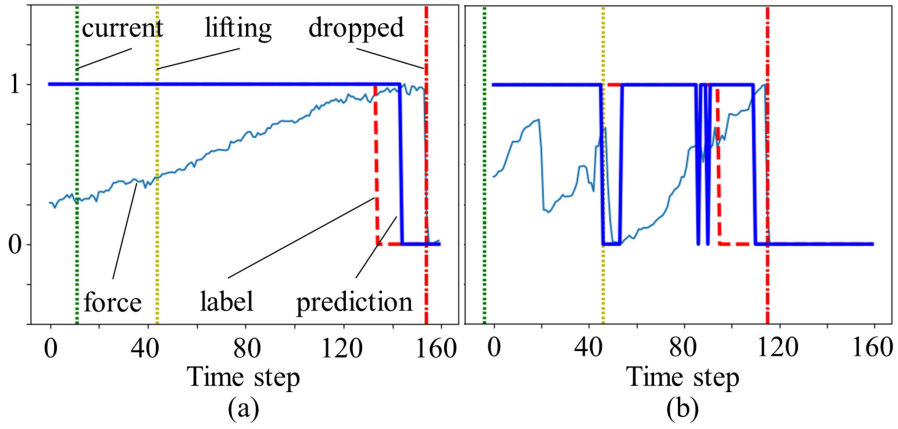


Figure 8.12: Prediction result of (Data + STFT) & LSTM model

In Figure 8.12 (a), the force changes significantly after the lifting time step, and the unstable state is hard to judge from force change. The prediction results closely matches the label data most of the time, and the predicted unstable point is ahead of dropping time. Since unstable state time is labeled 20 time steps before dropping time and cannot be precisely shown in the graph, the prediction data cannot reach

100% accuracy. However, this type of prediction can still be considered a good result.

Figure 8.12 (b) shows an obvious stability prediction at the lifting time step. As the force changes to 0, it can be assumed that an unstable state has occurred. However, upon reviewing the video, there was no obvious evidence of unstable grasping. This "wrong prediction" is a result of using single channel input. At that time, the robotic hand lifted the item and the forces of different fingers changed before reaching a new balance to achieve stable grasping. Only this channel witnessed this type of "unstable state", but it should still be considered a stable status.

Based on these partial results, there can get two points: 1) STFT data improves the success rate and enables more precise and stable predictions; 2) time data directly improves the ahead drop rate and is crucial for grasp stability prediction.

Compared with 84.60% in Table 8.1, training with a single channel results in a lower success rate, possibly because data from multiple channels can provide additional information, such as contact information between hand and item. Therefore, different data channels can be combined in specific use cases to improve the success rate.

8.4.3 Verification on Stable Grasping

The performance of four different models on DoraHand is shown in Table 8.5 and Figure 8.13. The success rate is similar to the result of the dataset, with LSTM and (Data + STFT) & LSTM showing good performance at over 85%. The low ahead drop rate may be caused by the differences in data acquisition methods. The grasp stability prediction was tested in a relatively real situation, where no unstable state occurred during most of the testing time. As a result, the ahead drop rate is not used for comparison in this case.

Table 8.5: Comparison of four models on DoraHand

Model	Success rate
LSTM	89.38%
STFT & LSTM	80.19%
(Data + STFT) & LSTM	87.24%
LSTM + STFT & LSTM	74.39%

When comparing the details of the prediction results, four different phases were labeled in Figure 8.13 (a): 1) adding force; 2) slipping; 3) holding the item; 4) dropping the item. LSTM shows the best success rate, as seen in Figure 8.13 (a), but the prediction result only shows the dropping time in phase 4, which does not meet the requirements for real-world applications. The other three models with STFT input predict stability result in phase 2 and output stable result in phase 3, demonstrating

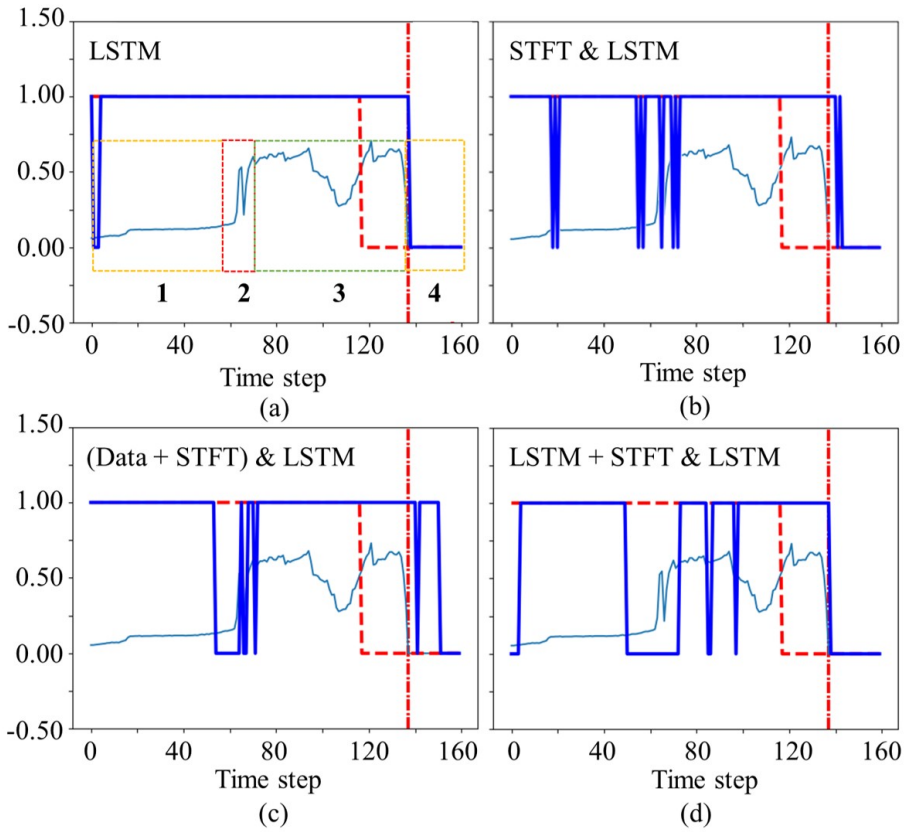


Figure 8.13: Results of the four models by DoraHand

their grasp stability prediction capability and highlighting the difference in force change between phase 2 and 3. This comparison shows that frequency data can help to predict unstable states through changes in force values.

To verify the performance of the stable grasping function, the hand was set to grasp the item with light force and tested in different situations, as shown in Figure 8.14. Experiments (a) and (b) were separately tested with both disable and enable stable grasping functions. All four tests use the (Data + STFT) & LSTM model.

The comparison of disabled and enabled stable grasping functions for experiments (a) and (b) are shown in Figure 8.15. When the stable grasping function was disabled, the item was dropped as shown in Figure 8.15 (a0) and Figure 8.15 (b0). When the stable grasping function was enabled, the item was lifted as shown in Figure 8.15 (a1) and Figure 8.15 (b1). The (a1) demonstrates that the stable grasping function can help to grasp an item with an unknown weight of 2 kg. The (b1) can react quickly enough to hold the item even if there is a sudden impact with a 2 kg weight. The experiment (c) in Figure 8.14 (c) becomes more stable as the weight is gradually increased. The experiment (d) in Figure 8.14 (d) shows the case that rotates the item inside the hand, the grasping becomes tighter after several rotations.

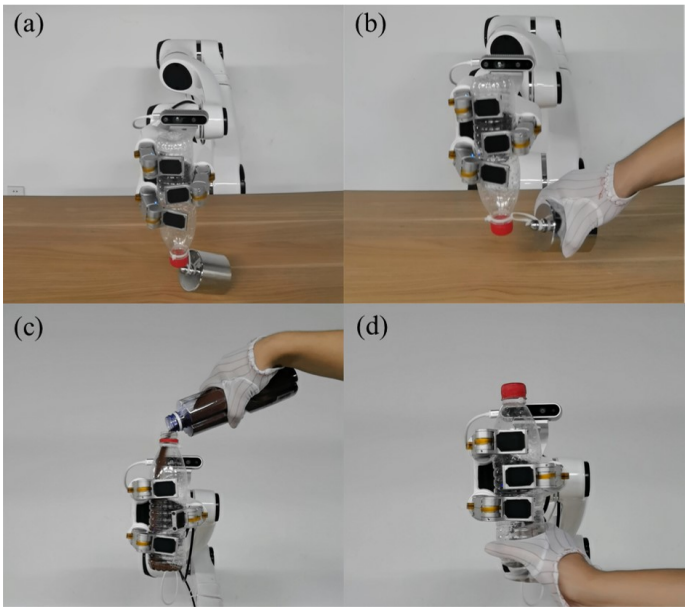


Figure 8.14: Four stable grasping test scenarios

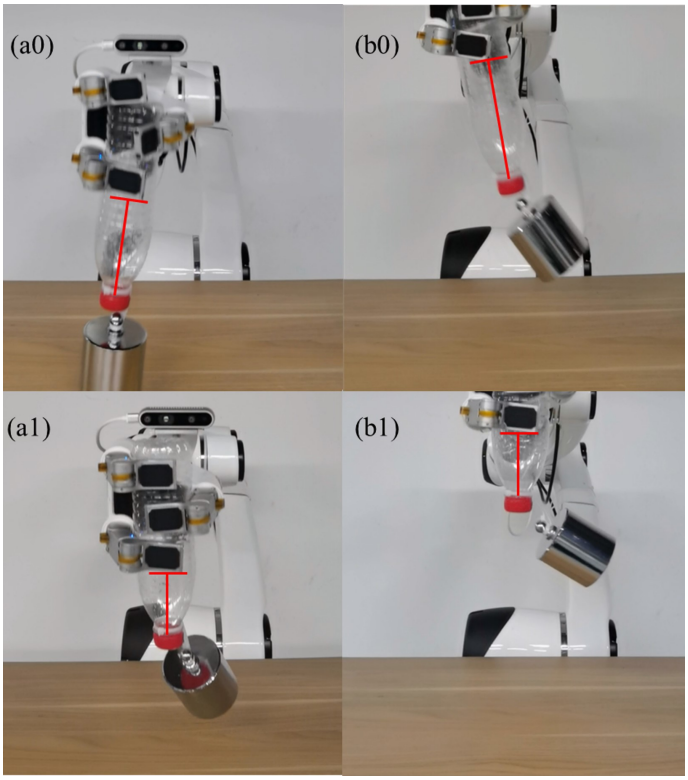


Figure 8.15: Results of experiments (a) and (b)

Force feedback was recorded during these experiments, and it can be easily observed that force increases with the prediction result, as shown in Figure 8.16. Figure 8.16 (a) shows the force change in experiment (b). It can be seen that force increases within 0.15 s, and a force drop just happened between 0.925 - 1.05 s. This means that the algorithm is fast enough to predict stability within 1 - 2 time frames of data. 1 time frame includes 16 channels and takes 64 ms for 16 predictions. Figure 8.16 (b) shows the force change in experiment (d), demonstrating that the force increases step by step after each prediction.

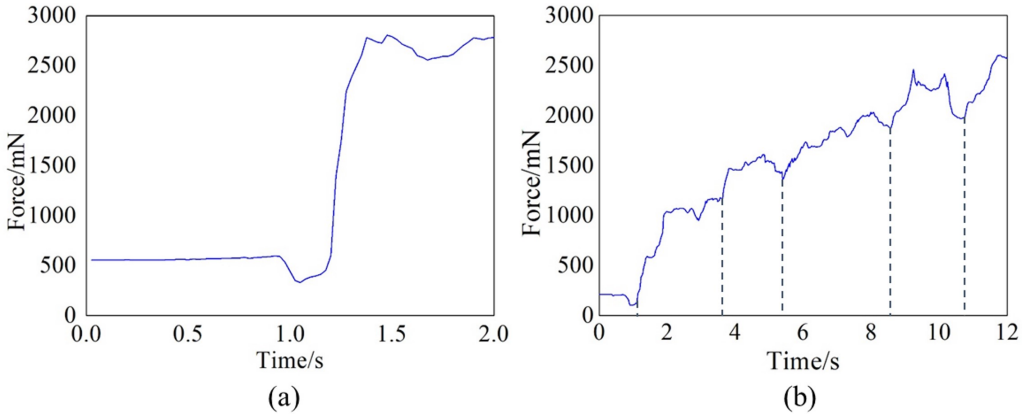


Figure 8.16: Tactile sensor data of experiments (b) and (d)

With this verification applied on the dexterous hand, three key points are demonstrated: 1) frequency data can help identify more slip features; 2) the labeling method is critical to the training process. A new dataset is needed to achieve more accurate results by labeling which stages matter for specific use cases. Labeling methods using other sensors, such as accelerometers, may be helpful [Al-Shanoon and Ahmad, 2015]; 3) the stable grasping function of the dexterous hand is already tested by DoraHand, the performance can be used in real use cases and ensure the stable grasping with unknown weight items and withstand certain external forces.

8.4.4 Generalization ability on Pressure Data

According to the designed experiment, data from the pressure sensors was selected as the input and separated into 316 batches, each with a size of 160 time steps. The results are shown in Table 8.6 and Figure 8.17.

Among the four models, (Data + STFT) & LSTM performs the best with the highest success rate in the 71 Hz test, demonstrating good generalization ability. All these models have lower performance in the 17.75 Hz test, possibly because the pressure data during the dropping process of the suction cup gripper changes much

Table 8.6: The experiment result of four models with suction cup gripper

Model	17.75 Hz	71 Hz
	Success rate	Success rate
LSTM	64.15%	78.69%
STFT & LSTM	57.11%	71.91%
(Data + STFT) & LSTM	58.25%	78.80%
LSTM + STFT & LSTM	60.11%	76.65%

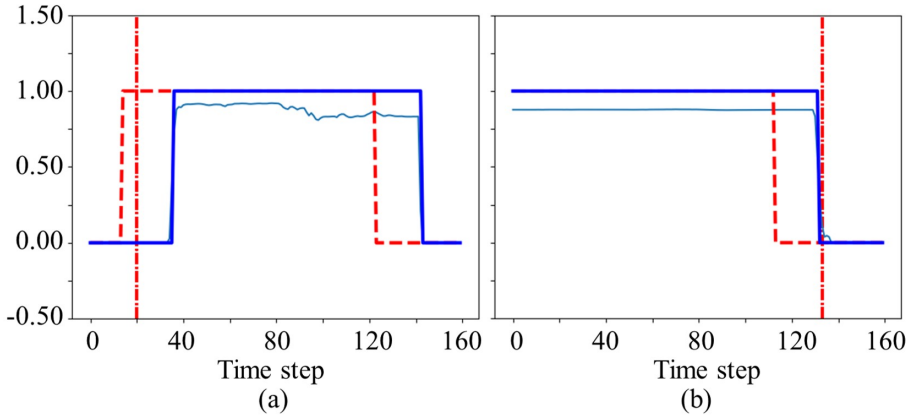


Figure 8.17: Results of (Data + STFT) & LSTM with suction cup gripper

faster than the tactile data of the dexterous hand.

As seen in Figure 8.17 (a), the output prediction result, represented by the blue bold line, mostly coincides with the labeled pressure data. However, the pressure changes too quickly to make a prediction ahead of the drop. The pressure value drops rapidly when there is a significant vibration or air leakage. The low success rate may be due to the fact that labeling 20 steps ahead of the drop for slip label is too many in this case, causing the success rate to decrease to 62.5%.

Figure 8.17 (b) shows a good prediction result by LSTM + STFT & LSTM on 71 Hz data, where the prediction result is located between unstable state label and drop label. However, the data still changes too quickly and is not sufficient for prediction. A model trained with higher frequency pressure data and better labeling methods may help achieve better results.

From this experiment, three key points can be summarized: 1) (Data + STFT) & LSTM show good capability in stability prediction and can be generalized to other time series data; 2) features in time series are easier to transfer to other targets than those in frequency domain; 3) higher frequency data can help to improve training and prediction results.

8.5 Conclusion

The grasp stability prediction function for stable grasping has been developed based on the visual-tactile dataset and verified through different experiments. A comparison of several potential models suggests that combining STFT and LSTM improves the success rate, and input data with a higher frequency significantly benefits the results.

The stable grasping function is verified with the DoraHand and demonstrated its capability in real-world use cases. Tests on the pressure sensor data have shown the generalization ability on time-series data. Data frequency significantly influences the result when using STFT, with the time domain performing better than the frequency domain in terms of generalization ability.

In terms of model details, the model of (Data + STFT) & LSTM improves the precision of grasp stability prediction and can continue to be improved in terms of ahead drop rate. More STFT & LSTM-based solutions for grasp stability prediction can be explored. These candidate models can be applied to more realistic situations, and the applications in tactile-based stable grasping can be further developed.

Part V

Conclusion and Outlook

Chapter 9

Conclusion and Outlook

This final chapter serves as a conclusion to the thesis, summarizing the work that has been done, presenting findings related to the research question, discussing the lessons learned throughout the thesis work, and providing insights for future work. The chapter is divided into four sections. Section 9.1 presents a summary of the entire thesis. Section 9.2 introduces some results obtained during the thesis preparation. Section 9.3 presents the key takeaways and lessons learned from this thesis work. Section 9.4 provides an outlook on the related technologies for future research.

9.1 Thesis Summary

The primary purpose of this thesis is to develop a novel modular dexterous hand for a robot grasping system. Stable grasping has been selected as the primary application case, as it is a key issue in robotic manipulation. The dexterous hands have been developed in two types and several versions, and the design of one product, DoraHand, is introduced with details. To make better use of the dexterous hand, visual-based grasp planning, and tactile-based stable grasping functions have been developed based on its capabilities. The work on grasp planning began with an analytic approach and was combined with a deep learning-based approach to achieve better results. The work on stable grasping started with a visual-tactile dataset, developed a grasp stability prediction solution using the combination of LSTM and STFT, and realized stable grasping with the DoraHand. This thesis covers these three main areas of work: dexterous hand, grasp planning, and stable grasping.

Dexterous hand The design of the dexterous hand involved two types and several versions, including the Eagle Shoal and DoraHand, with one key type of DoraHand detailed in this thesis. As a modular dexterous hand, its key features are the modular finger and tactile sensor module. With the modular finger, the

hand can be extended to different layouts, and two-finger and five-finger version have been verified and used in the applications. The tactile sensor module enables the dexterous hand's external sensing capability, which is essential in the field of robotic manipulation. The tactile sensor module also significantly reduces the cost of the DoraHand. The Eagle Shoal has won third place in manipulation competition with its manipulation capability [EagleShoal, 2017]. The DoraHand has been developed as a product and used by more than twenty research institutes and companies, winning product design awards including DIA, RedDot, and IF [Dorabot, 2020a, Dorabot, 2020b, Dorabot, 2021b, Dorabot, 2022]. This work developed a piratical dexterous hand that can help promote research in robotic manipulation.

Grasp planning The goal of grasp planning is to make better use of the dexterous hand and ensure its grasp capability. An analytic analysis has been conducted with the DoraHand and introduces the limitations of the dexterous hand layout to improve efficiency in the planning stage. As visual processing is an essential part of the entire system, an approach combining analytic and data-driven approaches has been developed with a new grasp plan representation for model output. The visual-based grasp planning has been verified with the DoraHand.

Stable grasping The goal of stable grasping is to act as a fundamental function of a dexterous hand in applications. With the idea of learning from data, a visual-tactile dataset has been built, and work based on this dataset has been developed. This visual-tactile dataset combines visual and tactile data and is open-source. The work on stable grasping based on this dataset has successfully realized the stable grasping function and demonstrated the generalization ability in pressure sensor data. With this work, the dexterous hand can realize the stable grasping function with tactile data.

The outputs of this work form a stable grasping system that can also serve as a platform for research related to robotic manipulation. The hardware and open-source data contribute to the related research areas.

9.2 Further Results

During the preparation of this thesis, work related to the thesis topic has been ongoing, including hardware and software applications.

Dexterous hand iteration With the increasing application of the dexterous hand, feedback from users has been instrumental in improving its design. The iteration

started with the Eagle Shoal, which was designed for service robots and gradually changed to the DoraHand, targeting industrial applications. The design has become more mature and the components more stable. During the preparation of this thesis, the DoraHand underwent seven iterations, resulting in new versions of two-finger and five-finger hands that are more suitable for real-world applications.

Tactile-based interaction By taking advantage of the tactile sensor, interaction between robot and human can be achieved. The grasp stability prediction used for the stable grasping function can be used to detect interactions in the human-robot interaction process. As humans interact with the hand, it can sense differences and classify them as target interaction commands, which can be a potential application in human-robot interaction.

9.3 Lessons Learned

In addition to achieving the main objectives of this thesis, general methodologies for the robotic grasping system have been identified and are presented below.

Ensuring the stability of the dexterous hand As a design that evolves from an initial idea to a product, the dexterous hand requires detailed design and procedures to ensure stable performance.

1. **Sensor calibration procedure**. Sensor calibration is a critical procedure for the dexterous hand. While there is a specific device designed for calibration, further detailed work is still needed in practice. During testing, creep situations in the sensors and impact forces can influence the results. Therefore, it is necessary to carefully control the procedure of adding the payload, and set suitable criteria for data fitting and calibration.

2. **FPC**. The FPC is used in this design. Although it may influence the motion flexibility of the joint, it is still a better solution during the exploration due to the balance between limited joint space and lifetime. The thickness, material, and cable layout of the FPC are also key points that can influence its lifetime.

3. **Integration with the robot arm**. Due to different integration requirements with different robot arms, a universal physical interface component is necessary for integration. Different solutions have been tested, and the current one with the screw connected in the radial direction can better fix the hand with precision and stability.

4. **Joint parameter calibration**. As the joint parameters may change during use, several thousand tests are necessary for the joint to reach a good status and ensure a stable hand. In case of parameters change in the future, a user-friendly CPA page has been designed to calibrate and change the parameters.

Quality of the dataset During the process of building the visual-tactile dataset, several points could be improved to enhance its quality, as outlined below.

1. **Sensor selection**. While the visual sensor used in the dataset has similar parameters to those used in normal settings, higher quality sensors would be necessary to improve the data quality, and the interference between different sensors also needs to be considered.

2. **Key point labeling**. Although automatic labeling is used in the dataset, some key points, such as unstable state or slip, are still missed. Additional sensors, such as acceleration sensors, could help provide more information and improve the dataset.

3. **Influence of object deformation**. After thousands of tests, the quality of objects may have changed and influenced the data quality. Therefore, preparing more samples with consistent quality would be necessary to guarantee data quality.

4. **Time synchronization**.Proper time synchronization is important for a dataset. The current solution uses the same length of time step, but a better method with the same timestamp and better synchronization with video data could improve the dataset.

System in real application While the experiments conducted in this thesis focused on specific targets, building a complete system for real-world applications involves several considerations that were not covered.

1. **Poor quality visual data**. In real-world applications, the quality of visual data is often poor, including RGB and depth data. Point clouds are particularly susceptible to poor data quality. Backup solutions, multi-view, or multi-test solutions are necessary to ensure reliable results.

2. **Robot arm motion planning**. Robot arm motion planning is a key aspect of real-world applications that was not introduced in this thesis. Collision avoidance and planning speed are critical parameters that can affect the success of the application.

3. **Uncertain command delays**. Delays are common in the real world and can affect the execution of tasks, especially for manipulation tasks. Estimating delays accurately and monitoring conditions can help improve task performance.

4. **Deformation of the target**. Real-world objects are not rigid bodies, and deformation can significantly affect manipulation tasks, particularly for grasping tasks that involve dexterous hand end-effectors or suction cup grippers. Simulating deformation is difficult, and therefore, the ability to adapt to changing conditions and evaluate grasp stability is crucial.

9.4 Outlook

This section introduces some potential directions that can be built upon the current system and improved in the next step.

More stable dexterous hand with diverse sensors In terms of dexterous hand development, there is still much room for improvement. Joint speed, motion precision, and sensor sensing capability can all be further improved. As visual-based tactile sensors continue to develop, it may be possible to develop a sensor within a thin volume. Integrating such sensors into the hand would be a valuable direction to pursue. For the manipulation capability of the hand, under-actuated joints mimic the DIP joint can be developed, and DOF in the palm can also be improved.

Higher quality dataset for robotic manipulation In addition to the work presented in this thesis, there is a growing need for large datasets in the field of robotics compared to the huge data in vision processing and NLP. High-quality data is particularly valuable, and there is potential for creating datasets that are as useful as those currently available in the visual field, where researchers can segment anything [Kirillov et al., 2023]. The development of such datasets would be an important step toward achieving high performance in robotics.

Learning-based solutions with task planner In addition to focusing on executing specific tasks, it is also valuable to consider how these tasks can be connected to create more complex procedures. Building on the stable grasping, it is possible to apply this work to complex manipulation tasks. Like chatting with humans, designing the task with available functions is a reasonable task for a learning-based solution [OpenAI, 2023]. One potential direction is to explore the use of task planner to connect individual actions and achieve more intelligent manipulation solutions.

As Artificial Intelligence (AI) continues to develop, there is great potential for growth in the field of robotics. As more work from the visual, speech, and NLP fields are applied to robotics, it is likely that more intelligent systems will be achieved. Iterative improvements in hardware and algorithms will help bridge the gap between the virtual and physical worlds.

List of Figures

List of Tables

Acronyms

2D	Two Dimensional
3D	Three Dimensional
AI	Artificial Intelligence
BLDC	BrushLess Direct Current
CAN	Controller Area Network
CNN	Convolutional Neural Network
ConvLSTM	Convolutional LSTM
CPA	Control Performance Assessment
DC	Direct Current
DIP	Distal Interphalangeal
DNN	Deep Neural Network
DOF	Degrees of Freedom
FC	Fully Connected
FPC	Flexible Printed Circuit
GFPR	Graspable Finger Position Regions
GQ-CNN	Grasp Quality Convolutional Neural Network
GWS	Grasp Wrench Space
ICRs	Independent Contact Regions
KIT	Karlsruhe Institute of Technology
KNN	K-Nearest Neighbors

LSTM	Long Short-Term Memory
MCP	Metacarpophalangeal
NB	Naive Bayes
NLP	Natural Language Processing
PID	Proportional Integral Derivative
PIP	Proximal Interphalangeal
RBF	Radial Basis Function
RGB	Red, Green, and Blue
RGBD	RGB and Depth
RNN	Recurrent Neural Network
ROS	Robot Operating System
SPI	Serial Peripheral Interface
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
TOF	Time-Of-Flight
UR	Universal Robots
URDF	Unified Robot Description Format
USB	Universal Serial Bus
VGG	Visual Geometry Group
VR	Virtual Reality
YCB	Yale-CMU-Berkeley

Bibliography

[Abadi et al., 2016] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation (OSDI)*, pages 265–283.

[Agur and Dalley, 2009] Agur, A. M. and Dalley, A. F. (2009). *Grant's atlas of anatomy*. Lippincott Williams & Wilkins.

[Akkaya et al., 2019] Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., Schneider, J., Tezak, N., Tworek, J., Welinder, P., Weng, L., Yuan, Q., Zaremba, W., and Zhang, L. (2019). Solving rubik's cube with a robot hand. *arXiv preprint*.

[Al-Shanoon and Ahmad, 2015] Al-Shanoon, A. A. S. and Ahmad, S. A. (2015). Slip detection with accelerometer and tactile sensors in a robotic hand model. *IOP Conference Series: Materials Science and Engineering*, 99.

[Alex, 2021] Alex, P. (2021). keypoint_rcnn_training_pytorch. https://github.com/alexppppp/keypoint_rcnn_training_pytorch. Accessed: 05.2023.

[Altman, 1992] Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46:175–185.

[ATI, 1989] ATI (1989). Ati sensor. https://www.ati-ia.com/index.aspx. Accessed: 05.2023.

[Bain et al., 2015] Bain, G. I., Polites, N., Higgs, B. G., Heptinstall, R. J., and McGrath, A. M. (2015). The functional range of motion of the finger joints. *Journal of Hand Surgery (European Volume)*, 40:406–411.

[Barrett, 2012] Barrett (2012). Barrett support. https://support.barrett.com/. Accessed: 05.2023.

[Bhirangi et al., 2021] Bhirangi, R., Hellebrekers, T., Majidi, C., and Gupta, A. (2021). Reskin: versatile, replaceable, lasting tactile skins. *arXiv preprint*.

[Boff and Lincoln, 1988] Boff, K. R. and Lincoln, J. E. (1988). *Engineering Data Compendium. Human Perception and Performance, Volume 1*. NASA.

[Bohg et al., 2013] Bohg, J., Morales, A., Asfour, T., and Kragic, D. (2013). Data-driven grasp synthesis - a survey. *IEEE Transactions on Robotics*, 30:289–309.

[Calli et al., 2015] Calli, B., Walsman, A., Singh, A., Srinivasa, S., Abbeel, P., and Dollar, A. M. (2015). Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set. *IEEE Robotics & Automation Magazine*, 22:36–52.

[Cambridge, 2023] Cambridge, D. (2023). dexterous. `https://dictionary.cambridge.org/`. Accessed: 05.2023.

[Cavallo et al., 2014] Cavallo, A., De Maria, G., Natale, C., and Pirozzi, S. (2014). Slipping detection and avoidance based on kalman filter. *Mechatronics*, 24:489–499.

[Chang et al., 2010] Chang, L. Y., Srinivasa, S. S., and Pollard, N. S. (2010). Planning pre-grasp manipulation for transport tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2697–2704.

[Chebotar et al., 2016a] Chebotar, Y., Hausman, K., Su, Z., Molchanov, A., Kroemer, O., Sukhatme, G., and Schaal, S. (2016a). Bigs: Biotac grasp stability dataset. In *ICRA 2016 Workshop on Grasping and Manipulation Datasets*, pages 1–8.

[Chebotar et al., 2016b] Chebotar, Y., Hausman, K., Su, Z., Sukhatme, G. S., and Schaal, S. (2016b). Self-supervised regrasping using spatio-temporal tactile features and reinforcement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1960–1966.

[Chen et al., 2018] Chen, W., Khamis, H., Birznieks, I., Lepora, N. F., and Redmond, S. J. (2018). Tactile sensors for friction estimation and incipient slip detection—toward dexterous robotic manipulation: A review. *IEEE Sensors Journal*, 18:9049–9064.

[Christopher, 2015] Christopher, O. (2015). Understanding lstm networks. `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`. Accessed: 05.2023.

[Cornelia and Suárez, 2005] Cornelia, J. and Suárez, R. (2005). Determining independent grasp regions on 2d discrete objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2941–2946.

[Cornella and Suárez, 2005] Cornella, J. and Suárez, R. (2005). Fast and flexible determination of force-closure independent regions to grasp polygonal objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 766–771.

[D. et al., 2009] D., G., Gorges, N., and Worn, H. (2009). Tactile sensing for an anthropomorphic robotic hand: hardware and signal processing. In *International Conference on Robotics and Automation (ICRA)*.

[Dahiya et al., 2009] Dahiya, R. S., Metta, G., Valle, M., and Sandini, G. (2009). Tactile sensing—from humans to humanoids. *IEEE transactions on robotics*, 26:1–20.

[Dahiya and Valle, 2008] Dahiya, R. S. and Valle, M. (2008). Tactile sensing for robotic applications. *Sensors, Focus on Tactile, Force and Stress Sensors*, pages 298–304.

[Deimel and Brock, 2016] Deimel, R. and Brock, O. (2016). A novel type of compliant and underactuated robotic hand for dexterous grasping. *International Journal of Robotics Research*, 35:161–185.

[Ding et al., 2001] Ding, D., Lee, Y. H., and Wang, S. (2001). Computation of 3-d form-closure grasps. *IEEE Transactions on Robotics and Automation*, 17:515–522.

[Dizioğlu and Lakshiminarayana, 1984] Dizioğlu, B. and Lakshiminarayana, K. (1984). Mechanics of form closure. *Acta mechanica*, 52:107–118.

[Dong et al., 2021] Dong, M., Wei, S., Yu, X., and Yin, J. (2021). Mask-gd segmentation based robotic grasp detection. *Computer Communications*, pages 124–130.

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

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

[Dorabot, 2020a] Dorabot (2020a). Dorahand. `https://dorabot.com.cn/Products/info_itemid_45.html`. Accessed: 05.2023.

[Dorabot, 2020b] Dorabot (2020b). Dorahanddia. `https://en.di-award.org/collections/detail/667.html?page_size%3D1000%26page%3D1%26year%3D2020%26award_type%3D%26award_group%3D1%26category_id%3D`. Accessed: 05.2023.

[Dorabot, 2021a] Dorabot (2021a). Dorahand. https://github.com/dorabot/DoraHand. Accessed: 05.2023.

[Dorabot, 2021b] Dorabot (2021b). Dorahandreddot. https://www.red-dot.org/project/dorahand-f3-51430. Accessed: 05.2023.

[Dorabot, 2022] Dorabot (2022). Dorahandif. https://ifdesign.com/en/winner-ranking/project/dorahand-3f/336442. Accessed: 05.2023.

[EagleShoal, 2017] EagleShoal (2017). 2nd robotic grasping and manipulation competition @ iros 2017. http://www.rhgm.org/activities/competition_iros2017/. Accessed: 05.2023.

[ElKoura and Singh, 2003] ElKoura, G. and Singh, K. (2003). Handrix: animating the human hand. In *ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 110–119.

[Farrow and Correll, 2015] Farrow, N. and Correll, N. (2015). A soft pneumatic actuator that can sense grasp and touch. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2317–2323.

[Faverjon and Ponce, 1991] Faverjon, B. and Ponce, J. (1991). On computing two-finger force-closure grasps of curved 2d objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 424–429.

[Fernandez et al., 2014] Fernandez, R., Payo, I., Vazquez, A., and Becedas, J. (2014). Micro-vibration-based slip detection in tactile force sensors. *Sensors*, 14:709–730.

[Ferrari and Canny, 1992] Ferrari, C. and Canny, J. F. (1992). Planning optimal grasps. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2290–2295.

[Goldfeder et al., 2009] Goldfeder, C., Ciocarlie, M., Dang, H., and Allen, P. K. (2009). The columbia grasp database. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1710–1716.

[Graves et al., 2013] Graves, A., Mohamed, A. R., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649.

[Guo et al., 2017] Guo, D., Sun, F., Liu, H., Kong, T., Fang, B., and Xi, N. (2017). A hybrid deep architecture for robotic grasp detection. In *IEEE International conference on robotics and automation (ICRA)*, pages 1609–1614.

[He et al., 2017] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *IEEE international conference on computer vision (ICCV)*, pages 2961–2969.

[Hearst et al., 1998] Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Schölkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications*, 13:18–28.

[Heyneman and Cutkosky, 2016] Heyneman, B. and Cutkosky, M. R. (2016). Slip classification for dynamic tactile array sensors. *The International Journal of Robotics Research*, 35:404–421.

[Hildebrandt et al., 2008] Hildebrandt, M., Albiez, J., and Kirchner, F. (2008). Computer-based control of deep-sea manipulators. In *OCEANS 2008-MTS/IEEE Kobe Techno-Ocean*, pages 1–6.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9:1735–1780.

[Jia, 2004] Jia, Y. B. (2004). Computation on parametric curves with an application in grasping. *The International Journal of Robotics Research*, 23:827–857.

[John and Langley, 2013] John, G. H. and Langley, P. (2013). Estimating continuous distributions in bayesian classifiers. *arXiv preprint*.

[Kasper et al., 2012] Kasper, A., Xue, Z., and Dillmann, R. (2012). The kit object models database: An object model database for object recognition, localization and manipulation in service robotics. *The International Journal of Robotics Research*, 31:927–934.

[Kawasaki et al., 2002] Kawasaki, H., Komatsu, T., and Uchiyama, K. (2002). Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand ii. *IEEE/ASME transactions on mechatronics*, 7:296–303.

[Kim et al., 2020] Kim, Y. J., Song, H., and Maeng, C. Y. (2020). Blt gripper: An adaptive gripper with active transition capability between precise pinch and compliant grasp. *IEEE Robotics and Automation Letters*, 5:5518–5525.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint*.

[Kirillov et al., 2023] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W. Y., Dollar, P., and Girshick, R. (2023). Segment anything. *arXiv preprint*.

[Kobayashi et al., 2012] Kobayashi, F., Kojima, F., Nakamoto, H., Kida, Y., Imamura, N., and Shirasawa, H. (2012). Slip detection with multi-axis force/torque sensor in universal robot hand. *International Journal of Applied Electromagnetics and Mechanics*, 39:1047–1054.

[Koenig and Howard, 2004] Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2149–2154.

[Lambeta et al., 2020] Lambeta, M., Chou, P. W., Tian, S., Yang, B., Maloon, B., Most, V. R., Stround, D., Santos, R., Byagowi, A., Kammerer, G., Jayaraman, D., and Calandra, R. (2020). Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5:3838–3845.

[Levine et al., 2018] Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37:421–436.

[Li et al., 2018] Li, J., Dong, S., and Adelson, E. (2018). Slip detection with combined tactile and visual information. In *IEEE international conference on robotics and automation (ICRA)*, pages 7772–7777.

[Li et al., 2003] Li, J. W., Liu, H., and Cai, H. G. (2003). On computing three-finger force-closure grasps of 2-d and 3-d objects. *IEEE Transactions on Robotics and Automation*, 19:155–161.

[Li et al., 2002] Li, Y., Yu, Y., and Tsujio, S. (2002). An analytical grasp planning on given object with multifingered hand. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3749–3754.

[Liow et al., 2019] Liow, L., Clark, A. B., and Rojas, N. (2019). Olympic a modular, tendon-driven prosthetic hand with novel finger and wrist coupling mechanisms. *IEEE Robotics and Automation Letters*, 5:299–306.

[Liu et al., 2008] Liu, H., Meusel, P., Hirzinger, G., and Xie, Z. (2008). The modular multisensory dlr-hit-hand: Hardware and software architecture. In *Proceedings of ASTRA, 9th ESA Workshop on Advanced Space Technologies for Robotic and Automation*, pages 461–469.

[Liu et al., 2018] Liu, H., Wang, F., Sun, F., and Fang, B. (2018). Surface material retrieval using weakly paired cross-modal learning. *IEEE Transactions on Automation Science and Engineering*, 16:781–791.

[Liu, 2000] Liu, Y. H. (2000). Computing n-finger form-closure grasps on polygonal objects. *The International Journal of Robotics Research*, 19:149–158.

[Ma et al., 2013a] Ma, R. R., Odhner, L. U., and Dollar, A. M. (2013a). A modular, open-source 3d printed underactuated hand. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2737–2743.

[Ma et al., 2013b] Ma, R. R., Odhner, L. U., and Dollar, A. M. (2013b). A modular, open-source 3d printed underactuated hand. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2737–2743.

[Mahler et al., 2017] Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Ojea, J. A., and Goldberg, K. (2017). Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metric. *arXiv preprint*.

[Mahler et al., 2018] Mahler, J., Matl, M., Liu, X., Li, A., Gealy, D., and Goldberg, K. (2018). Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning. In *IEEE international conference on robotics and automation (ICRA)*, pages 5620–5627.

[Mahler et al., 2016] Mahler, J., Pokorny, F. T., Hou, B., Roderick, M., Laskey, M., Aubry, M., Kohlhoff, K., Kröger, T., Kuffner, J., and Goldberg, K. (2016). Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In *IEEE international conference on robotics and automation (ICRA)*, pages 1957–1964.

[Markenscoff and Papadimitriou, 1989] Markenscoff, X. and Papadimitriou, C. H. (1989). Optimum grip of a polygon. *The International Journal of Robotics Research*, 8:17–29.

[Massa and Girshick, 2018] Massa, F. and Girshick, R. (2018). maskrcnn-benchmark: Fast, modular reference implementation of instance segmentation and object detection algorithms in pytorch. `https://github.com/facebookresearch/maskrcnn-benchmark`. Accessed: 05.2023.

[Miller and Allen, 2004] Miller, A. T. and Allen, P. K. (2004). Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11:110–122.

[Nguyen, 1988] Nguyen, V. D. (1988). Constructing force-closure grasps. *The International Journal of Robotics Research*, 7:3–16.

[Niparnan and Sudsang, 2006] Niparnan, N. and Sudsang, A. (2006). Computing all force-closure grasps of 2d objects from contact point set. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1599–1604.

[OpenAI, 2023] OpenAI (2023). Gpt-4 technical report. *arXiv preprint*.

[Pinto and Gupta, 2016] Pinto, L. and Gupta, A. (2016). Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413.

[Pollard, 2004] Pollard, N. S. (2004). Closure and quality equivalence for efficient synthesis of grasps from examples. *The International Journal of Robotics Research*, 23:595–613.

[Ponce and Faverjon, 1995] Ponce, J. and Faverjon, B. (1995). On computing three-finger force-closure grasps of polygonal objects. *IEEE Transactions on robotics and automation*, 11:868–881.

[Ponce et al., 1997] Ponce, J., Sullivan, S., Sudsang, A., Boissonnat, J. D., and Merlet, J. P. (1997). On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *The International Journal of Robotics Research*, 16:11–35.

[Pybullet, 2016] Pybullet (2016). Pybullet. https://pybullet.org/wordpress/. Accessed: 05.2023.

[Quigley et al., 2014] Quigley, M., Salisbur, C., Ng, A. Y., and Salisbury, J. K. (2014). Mechatronic design of an integrated robotic hand. *The International Journal of Robotics Research*, 33:706–720.

[Roa and Suárez, 2008] Roa, M. A. and Suárez, R. (2008). Independent contact regions for frictional grasps on 3d objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1622–1627.

[Roa and Suárez, 2009] Roa, M. A. and Suárez, R. (2009). Computation of independent contact regions for grasping 3-d objects. *IEEE Transactions on Robotics and Automation*, 25:839–850.

[ROBOTICS, 2012] ROBOTICS, W. (2012). Allegro hand wiki. http://wiki.wonikrobotics.com/AllegroHandWiki/index.php/Allegro_Hand. Accessed: 05.2023.

[Robotiq, 2008] Robotiq (2008). 3-finger adaptive robot gripper. https://robotiq.com/products/3-finger-adaptive-robot-gripper. Accessed: 05.2023.

[Rohmer et al., 2013] Rohmer, E., Singh, S. P., and Freese, M. (2013). V-rep: A versatile and scalable robot simulation framework. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1321–1326.

[S. et al., 2019] S., Z.-I. B., Gil, P., and Torres, F. (2019). Learning spatio temporal tactile features with a convlstm for the direction of slip detection. *Sensors*, 19.

[Sak et al., 2014] Sak, H., Senior, A. W., and Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *arXiv preprint*.

[Salehian et al., 2016] Salehian, S. S. M., Khoramshahi, M., and Billard, A. (2016). A dynamical system approach for softly catching a flying object: Theory and experiment. *IEEE Transactions on Robotics*, 32:462–471.

[Schunk, 2008] Schunk (2008). Sdh servo-electric 3-finger gripping hand. `http://www.schunk-modular-robotics.com/en/home/products/servo-electric-3-finger-gripping-hand-sdh.html`. Accessed: 05.2022.

[Sermanet et al., 2018] Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., and Levine, S. (2018). Time-contrastive networks: Self-supervised learning from video. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141.

[ShadowRobotCompany, 2013] ShadowRobotCompany (2013). Shadow dexterous hand. `https://www.shadowrobot.com/dexterous-hand-series/`. Accessed: 05.2023.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint*.

[Smith et al., 1999] Smith, G., Lee, E., Goldberg, K., Bohringer, K., and Craig, J. (1999). Computing parallel-jaw grips. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1897–1903.

[Stachowsky et al., 2016] Stachowsky, M., Hummel, T., Moussa, M., and Abdullah, H. A. (2016). A slip detection and correction strategy for precision robot grasping. *IEEE/ASME Transactions on Mechatronics*, 21:2214–2226.

[Su et al., 2012] Su, Z., Fishel, J. A., Yamamoto, T., and Loeb, G. E. (2012). Use of tactile feedback to control exploratory movements to characterize object compliance. *Frontiers in neurorobotics*, 6:1–9.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*.

[SynTouch, 2007] SynTouch (2007). Biotac sensor. https://syntouchinc.com/robotics/. Accessed: 05.2023.

[Tenzer et al., 2014] Tenzer, Y., Jentoft, L. P., and Howe, R. D. (2014). The feel of mems barometers: Inexpensive and easily customized tactile array sensors. *IEEE Robotics & Automation Magazine*, 21:89–95.

[Townsend, 2000] Townsend, W. (2000). Barretthand grasper: Programmably flexible part handling and assembly. *Industrial Robot*, 27:181–188.

[Van Wyk and Falco, 2018] Van Wyk, K. and Falco, J. (2018). Calibration and analysis of tactile sensors as slip detectors. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2744–2751.

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

[Wall et al., 2017] Wall, V., Zöller, G., and Brock, O. (2017). A method for sensorizing soft actuators and its application to the rbo hand 2. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4965–4970.

[Wang, 2018] Wang, T. (2018). dataset. https://github.com/bravebin/Visual-Tactile_Dataset. Accessed: 05.2023.

[Wang et al., 2019a] Wang, T., Geng, Z., Kang, B., and Luo, X. (2019a). Eagle shoal: A new designed modular tactile sensing dexterous hand for domestic service robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 9087–9093.

[Wang and Kirchner, 2021] Wang, T. and Kirchner, F. (2021). Grasp stability prediction with time series data based on stft and lstm. *arXiv preprint*.

[Wang et al., 2022] Wang, T., Xie, Z., Li, Y., Zhang, Y., Zhang, H., and Kirchner, F. (2022). Dorahand a novel dexterous hand with tactile sensing finger module. *Industrial Robot*.

[Wang et al., 2019b] Wang, T., Yang, C., Kirchner, F., Du, P., Sun, F., and Fang, B. (2019b). Multimodal grasp data set: A novel visual–tactile data set for robotic manipulation. *International Journal of Advanced Robotic Systems*, 16.

[Xu et al., 2021] Xu, Z., Qi, B., Agrawal, S., and Song, S. (2021). Adagrasp: Learning an adaptive gripper-aware grasping policy. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4620–4626.

[Yan et al., 2022] Yan, J., Xu, Z., Shi, P., and Zhao, J. (2022). A human-inspired soft finger with dual-mode morphing enabled by variable stiffness mechanism. *Soft Robotics*, 9:399–411.

[Yousef et al., 2011] Yousef, H., Boukallel, M., and Althoefer, K. (2011). Tactile sensing for dexterous in-hand manipulation in robotics—a review. *Sensors and Actuators A: Physical*, 167:171–187.

[Yuan et al., 2020] Yuan, S., Epps, A. D., Nowak, J. B., and Salisbury, J. K. (2020). Design of a roller-based dexterous hand for object grasping and within-hand manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 8870–8876.

[Yuan et al., 2017] Yuan, W., Dong, S., and Adelson, E. H. (2017). Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17:2762.

[Zhang et al., 2022] Zhang, H., Tang, J., Sun, S., and Lan, X. (2022). Robotic grasping from classical to modern: A survey. *arXiv preprint*.

[Zhang et al., 2018] Zhang, T., McCarthy, Z., Jow, O., Lee, D., Chen, X., Goldberg, K., and Abbeel, P. (2018). Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635.

[Zhang et al., 2016] Zhang, Y., Duan, X. G., Zhong, G., and Deng, H. (2016). Initial slip detection and its application in biomimetic robotic hands. *IEEE Sensors Journal*, 16:7073–7080.

[Zhu and Wang, 2003] Zhu, X. and Wang, J. (2003). Synthesis of force-closure grasps on 3-d objects based on the q distance. *IEEE Transactions on Robotics and Automation*, 19:669–679.