# Region-Based Planning for a 3D In-Hand Manipulation Platform Leveraging Variable Friction Fingers and External Surfaces

by

Alp Sahin

M.S. Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Robotics Engineering

by

---

May 2021

APPROVED:

---

Professor Berk Calli, Thesis Advisor

---

Professor Jing Xiao

---

Professor Xiangrui Zeng

---

Professor Yunus Dogan Telliel

**Abstract**

Existing robotic in-hand manipulation approaches generally utilize complex hand designs or extrinsic dexterity based strategies via dynamic models of contact. Still, they can only accomplish limited object motion. This work combines a simple robot hand with variable friction fingers and extrinsic dexterity based strategies to achieve 3D in-hand manipulation. Leveraging variable friction mechanism, we implement manipulation actions (i.e. within-hand sliding, rotation and pivoting) in a quasi-static manner, using kinematic level formulations. For a suitable representation of daily manipulation tasks, a region-based in-hand manipulation problem is proposed. An in-hand manipulation planner utilizing an A* algorithm with a novel region-based heuristic is developed to solve for a sequence of manipulation actions that navigate the contact patches between the hand and the object surface to desired regions. A wide range of goal regions can be achieved via this approach, which is demonstrated through simulated and real robot experiments following a standardized in-hand manipulation benchmarking protocol.

## Acknowledgements

# Contents


**1 Introduction** **1**

**2 Related Work** **6**

2.1 In-Hand Manipulation Mechanisms . . . . . 6

2.2 Extrinsic Dexterity Based Approaches . . . . . 9

2.3 In-Hand Manipulation Planning and Control . . . . . 12

**3 3D In-Hand Manipulation** **18**

3.1 Variable Friction Hand . . . . . 19

3.1.1 In-Hand Sliding . . . . . 20

3.1.2 In-Hand Rotation . . . . . 21

3.1.3 Kinematics Model . . . . . 22

3.2 Extrinsic Dexterity Based Strategies . . . . . 22

3.2.1 Moving Contact Up and Down . . . . . 23

3.2.2 Pivoting . . . . . 24

**4 Region-Based Manipulation Planning** **27**

4.1 Problem Formulation . . . . . 27

4.2 A* Search Algorithm . . . . . 28

4.2.1 States . . . . . 29

4.2.2 Actions . . . 29
4.2.3 Transition Model . . . 30
4.2.4 Termination . . . 32
4.2.5 Cost Function . . . 33

**5 Experiments and Results** **39**
5.1 Simulated Experiments . . . 39
5.1.1 Variable Friction Plugin . . . 39
5.1.2 Experimental Framework for Simulation Parameters . . . 40
5.1.3 Simulation Outcomes . . . 41
5.2 Real-Robot Experiments . . . 41
5.2.1 Experimental Setup . . . 41
5.2.2 Results . . . 44

**6 Discussion and Future Work** **47**

**7 Broader Impacts** **50**

**8 Conclusion** **55**

# List of Figures

1.1 Proposed region-based motion planning algorithm generates a sequence of manipulation primitives, given the object geometry and goal regions. 3D manipulation setup makes use of controlled sliding and rotation via variable friction fingers and extrinsic dexterity based manipulation strategies. . . . . 4

3.1 Finger equipped with a friction switching mechanism . . . . . 19

3.2 In-hand sliding using variable friction fingers. . . . . 21

3.3 In-hand rotation using variable friction fingers. . . . . 22

3.4 (a) In prior work, a 2-DOF robot hand was proposed that is capable of translation and rotation in 2D. (b) By combining this hand with a robotic arm, extrinsic dexterity based manipulation primitives such as prehensile pushing and pivoting can be utilized to achieve 3D manipulation. . . . . 23

3.5 Frames are attached on the robot end-effector, gripper palm, finger contact, object and pivoting center. Parameter $d_1$ is a fixed offset depending on the arm-gripper setup. Planning algorithm outputs the current finger parameters ($\theta_{finger}$, $d_2$, $d_3$ and $d_4$) for a pivoting action, which are then used in the kinematic model. During the pivoting, finger parameters are held constant, while the angles of the virtual joints ($\theta_{contact}$ and $\theta_{pivot}$) vary. . . . . 25

3.6 Pivoting is executed in two stages. At Stage 1, fingers are in high friction state, as the object rotates around the pivoting axis. At Stage 2, object is simultaneously rotated around the pivoting axis and finger contact by utilizing the table surface and switching to low friction state. . . . . 26

4.1 Pivoting and rotation resolutions for a hexagonal prism. The invalid orientation results in edge contacts when rotated, hence it is an invalid state. . . . . 30

4.2 When an action is taken, contact regions are transformed according to a transition model. Following an action, the contact region can translate and rotate on the same surface or travel between different surfaces. . . . . 31

4.3 This figure illustrates the changes in contact regions for left and right fingers for each motion primitive on a square prism. . . . . 33

4.4 To compute the heuristic function, surfaces of the object are unfolded, while keeping the contact surface at the origin. For each goal region the shortest distances with the contact are computed. Minimum of the shortest distances is the heuristic for the selected finger. . . . . 36

5.1 Simulation environment for 3D in-hand manipulation platform including variable friction gripper and Franka Emika Panda arm. . . . . 40
5.2 User interface for runtime simulation parameter adjustments and high-level in-hand manipulation commands. . . . . 42
5.3 Real robot experimental setup including the variable friction gripper and Franka Emika Panda arm. . . . . 43
5.4 Artificial objects are modeled and 3D-printed for the experiments. Set of objects include: 1) square prism 2) rectangular prism curved 3) rectangular prism large 4) hexagonal prism tall 5) rectangular prism small 6) hexagonal prism short. . . . . 43
5.5 Initial and goal region pairs are defined on each object. Red regions indicate the initial regions and blue regions indicate the goal regions. 44
5.6 Distribution of averaged goal region overlap for left and right fingers on each object. Results include 11 successful trials for square prism, 9 successful trials for rectangular prism curved and rectangular prism large, 5 successful trials for hexagonal prism tall, hexagonal prism short and rectangular prism small for a total of 44 successful trials. . 45

# List of Tables

3.1 In-hand sliding primitives . . . . . 20
3.2 In-hand rotation primitives . . . . . 21
3.3 DH Parameters for arm-hand system . . . . . 24

4.1 Effects of each primitive on system state parameters. NE denotes no effect. . . . . 32
4.2 Action cost multipliers for each primitive category. . . . . 34

5.1 Experiment Metrics . . . . . 44

# Chapter 1

# Introduction

In-hand manipulation (IHM), also called dexterous manipulation, refers to the changing of the grasp of an object within the hand without the need for regrasping [1]. Humans perform in-hand manipulation before, during, and after many daily tasks such as writing with a pen or using a key to unlock. Having dexterous skills, humans can use the tools that are picked up from pockets and tables, and avoid fatigue due to frequent arm motion.

In robotics, dexterous manipulation considers the cooperation of multiple manipulators or fingers for grasping and manipulating objects [2]. For robots, in-hand manipulation capabilities serve several purposes. Robots with these capabilities can work around their singularities and workspace limits, reduce energy consumption by avoiding large arm motions, and perform tasks that require finer manipulation [3]. Existing research efforts in robotics aim to match and surpass human dexter-

This thesis is partially based on a manuscript that will be published in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) in 2021.

The work in this thesis was supported in part by the National Science Foundation under grant IIS-1900953.

The author of this thesis is a trainee in Future of Robots in the Workplace – Research & Development (FORW-RD) NRT Program.

ity. They address the robotic in-hand manipulation problem from several aspects including the mechanical design, sensing, control and planning.

A wide range of mechanical designs between simple 1-degree-of-freedom(DOF) parallel jaw grippers [4, 5, 6] and multifingered anthropomorphic hands with high-DOF [7, 8, 9] has been deployed for in-hand manipulation. Advanced hand designs exhibit adequate intrinsic dexterity, whereas simple grippers have to leverage external surfaces, inertial and gravitational effects to achieve extrinsic dexterity[10, 11, 12].

Researchers frequently use multi-modal sensing, complex control and planning schemes for high-DOF hands [13, 14]. Grippers that leverage external factors such as inertia, gravity, and friction usually require knowledge about mechanical properties and dynamic models[15, 11]. In contrast, compliant and adaptive hands can perform in-hand manipulation without requiring complex sensing and modeling[16, 17].

In this thesis, we use the variable friction gripper introduced in [18], a 2-DOF hand design with active surfaces, that enable friction modulation. This design is capable of controlled sliding and rotation within-hand in a quasi-static manner through kinematics formulations [19]. However, the object motion with this hand is limited to a 2D plane. By leveraging the friction modulation, we implement extrinsic dexterity based manipulation strategies in kinematic level to extend the manipulation workspace to 3D. Resulting 3D in-hand manipulation platform works with geometric object models in an open loop manner without requiring any force sensing or control. We consider a region-based manipulation task to achieve desired contact regions on the object surface and propose a heuristic-based planning algorithm for region-based in-hand manipulation. We believe this planner can easily be adapted to similar hand designs [20].

This work's contributions are as follows (summarized in Figure 1):

- **A novel region-based WIHM method:** Different from the existing algorithms in literature, which formulate WIHM with either object pose references or desired point contacts, our planner works with contact areas and desired contact regions. We believe that this is not only a more realistic implementation for soft and/or flat finger surfaces (which are very common)[20, 16, 17], but also a useful formulation for achieving various robotic tasks. For example, as humans when we grasp a key and re-position it within our hand, we do not have desired point locations that we want to achieve on the key. Rather, we shift our fingers to a particular region on the key, while leaving the tip region free. The proposed planner utilizes a very similar formulation.

- **Quasi-static extrinsic dexterity:** We leverage the ability of being able to alter the friction at the contact locations for achieving extrinsic dexterity-based actions. Where the existing works on extrinsic dexterity based techniques mostly rely on dynamic models and knowledge of mechanical properties, we can reliably pivot and slide the object within hand using extrinsic contacts via kinematics-level formulations, without requiring any force sensing or control, thanks to the variable friction mechanism.

By combining these two strategies, we are able to conduct automatic WIHM in 3D, i.e. move the fingers from an initial contact region to a given desired region. To the best of our knowledge, the proposed system surpasses the automatic WIHM abilities of any other system in the literature: the planner can utilize controlled sliding, rotation and pivoting action and achieve trajectories with significantly large region displacements. We demonstrate these abilities in simulation and with real-robot experiments using a Franka Emika robot in Chapter 5. In this work, we assume that 3D geometry of the objects is known and the generated plans are

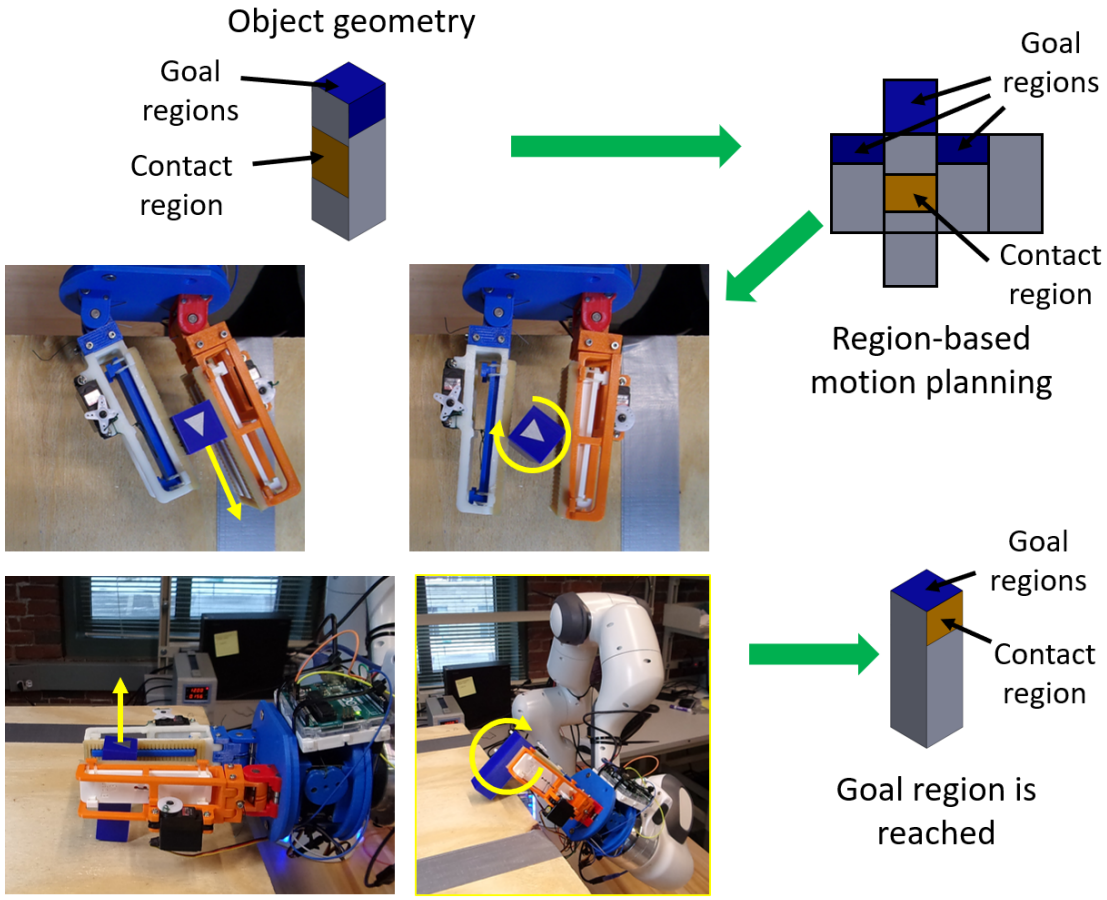


Figure 1.1: Proposed region-based motion planning algorithm generates a sequence of manipulation primitives, given the object geometry and goal regions. 3D manipulation setup makes use of controlled sliding and rotation via variable friction fingers and extrinsic dexterity based manipulation strategies.

executed in an open-loop manner. We discuss our future work regarding closed-loop implementations and possible mechanical improvements in Chapter 6.

# Chapter 2

# Related Work

In this chapter, we discuss the works in the literature that are related to the scope of this thesis. We start with reviewing the existing mechanisms and hand designs developed for in-hand manipulation. Then, we introduce the extrinsic dexterity based manipulation approaches. Finally, we provide an overview of different planning and control strategies for in-hand manipulation.

## 2.1 In-Hand Manipulation Mechanisms

Inspired by the dexterity of the human hand, researchers have developed anthropomorphic hands to perform complex in-hand manipulation tasks. These hands usually have multiple fingers, each with high degrees-of-freedom. The earliest designs include the Stanford/JPL Salisbury Hand [21] and the Utah/MIT Dexterous Hand [22]. These are 3 and 4 fingered designs, that include opposing thumbs resembling the structure of the human hand. Anthropomorphic hand designs in the literature utilize different actuation, drive, and sensing mechanisms. The most common approach is to use pulleys and tendons as a drive mechanism, and electric actuators for either fully-actuated[23] or underactuated[24, 25, 26, 27] systems. The Robonaut 2

Hand[28] includes a combination of both fully- and underactuated fingers. There are also gear-driven mechanisms as in [29, 13]. Unlike many others, hand designs in [30, 31] utilize pneumatic actuators. Whereas most of the designs consist of inflexible links made of rigid materials, there are also soft anthropomorphic hands with flexible fingers[31, 32]. Anthropomorhic hands are usually equipped with sensing mechanisms, starting with joint encoders and scaling up to force/torque sensors and tactile arrays on finger tips. The main challenge of using anthropomorphic hands for in-hand manipulation is the complexity and cost associated with modeling, sensing, and control, as these hands require accurate models and measurements for precise control to perform manipulation tasks.

Considering the requirements for different types of manipulation tasks, other multifingered hand designs are developed that address task-specific finger characteristics. These designs usually alter the configuration of the fingers deviating from the human hand structure[33, 34, 35], use bars and linkages as fingers[36, 37], or use parallel platforms instead of a fingered design for manipulation[38].

Alongside with the anthropomorphic and multifingered designs, researchers developed designs with lower degrees-of-freedom that leverage underactuated, compliant and adaptive mechanisms, fingers with active surfaces, and flexible joints. Following is the discussion of related examples in the literature.

In [39], Odhner and Dollar study the kinematics of underactuated manipulation to demonstrate the dexterous manipulation capabilities that can be achieved with elastic underactuated hands. This study led to several underactuated hand designs including [40, 41, 16]. In [40] Ma and Dollar propose a four-finger underactuated hand with flexible finger joints. The hand is capable of transferring objects from a pinch grasp to power grasp and rotating them around the hand normal. A two-finger design is proposed in [41], that utilizes a stationary thumb and a 2-DOF forefinger.

Using underactuated and fully-actuated approaches for controlling the forefinger, object sliding is performed between fingers. The GR2 Gripper is proposed in [16], which is an underactuated two-fingered hand with linkage based fingers that can reorient objects within hand. A search for optimized parameters for the linkage mechanism is provided in [42].

In [4], Chavan-Dafle et al. propose a 1-DOF parallel jaw gripper design utilizing elastic strips mounted over finger cavities to achieve contact geometry modulation between a free spinning point contact and a firm multipoint contact. By switching between these gripper modes, developed mechanism is capable of pivoting objects within the gripper considering a dynamic model of pivoting mechanics.

Similar to the GR2 gripper, an underactuated two-finger gripper with parallelogram linkages is proposed in [43]. In contrast to GR2 the linkages of this gripper are optimized to avoid parasitic rotations of the object, only allowing repositioning tasks within the hand.

In [44], Ward-Cherrier et al. utilize a novel TacTip tactile sensor at the fingertips of the GR2 gripper design to achieve model-free precise manipulation within-hand. Using Bayesian inference on tactile and visual data, active tactile manipulation is performed on cylindrical objects.

The JamHand[45], is a two-fingered hand equipped with pockets that store granular material at fingertips. The stiffness of the fingertips are modulated via the pressured air within the pockets. Primitives such as rolling, sliding, and finger-gaiting are achieved with minimal actuation.

Designs that incorporate active surfaces can actively change contact modes and forces during manipulation or actuate the surfaces for manipulating the objects directly. Velvet fingers proposed in [46], are equipped with conveyor belts that are capable of changing the surface friction of the fingers, pulling and pushing the

objects in and out of grasp, and rotation within hand. Ma and Dollar propose a two-fingered gripper with active surfaces in [41], where they utilize a thumb design equipped with a conveyor belt and a forefinger with passive rollers. This hand is capable of pushing and pulling objects within grasp, as well as in-hand alignment. Similarly, in [47], Govindan and Thondiyath designed a gripper that utilizes movable belts and spring loaded mechanisms to achieve contact force adjustments and shape conformity for the grasped objects. Fingers with actively driven rollers are used in [48] for 6-DOF spatial manipulation. In [20], a different variable friction mechanism is proposed, that is based on origami inspired surfaces. The surfaces can be folded and unfolded by a tendon driven mechanism thanks to the origami-based design. Unfolded and folded modes of the surface correspond to different friction states, enabling in-hand sliding and rotation.

The advantages of the compliant and adaptive designs is that they enable manipulation of a wide range of objects without the need for advanced sensing, and control schemes. They provide a robust approach for executing manipulation primitives such as in-hand sliding, rotation, pivoting. However, they leverage mechanical phenomena that is more difficult to model than the kinematics of a design with rigid fingers and revolute joints. This leads to challenges in planning for the in-hand manipulation.

## 2.2 Extrinsic Dexterity Based Approaches

Robotic hand and gripper designs that facilitate in-hand manipulation are discussed in Section 2.1. These mechanisms exhibit intrinsic dexterity, as they are capable of performing a wide range of manipulation primitives via their actuation and driving systems without relying on any external effects. However, these designs do not com-

pose the only solution to the in-hand manipulation problem. Researchers frequently use simple 1-DOF parallel jaw grippers and utilize extrinsic factors to compensate for the lack of dexterity of these mechanisms. The external factors include gravity, inertial effects, and surfaces or objects in the manipulation environment.

In [10], Chavan-Dafle et al. use a three-fingered simple robotic gripper that is only capable of closing on the objects. They implement extrinsic dexterity based manipulation primitives such as throw to palm, roll to ground, throw and flip, etc. in an open loop manner. In this conceptual work, they validate the extrinsic dexterity based in-hand manipulation framework. They show that it is possible to switch between different types of grasp using these primitives.

A simple parallel jaw gripper is used in [12] together with external surfaces in the manipulation environment to accomplish straight sliding, pivoting, and rolling. This work derives the mechanics of the interaction between gripper, object, and the environment. Chavan-Dafle and Rodriguez develop a quasi-dynamic formulation based on complementarity constraints to determine the forces required to achieve the desired object motion. In the future work [49], authors propose a dynamics representation called motion cones to describe the extrinsic dexterity based interactions between gripper, object, and the environment assuming known dynamic models.

In [11], Cruciani and Smith use a parallel jaw gripper connected to a robotic arm and utilize the inertial effects caused by the arm motion to perform object pivoting. Their strategy is based on low precision dynamic models of frictional and inertial effects, and a low quality camera that takes stationary images in between system actions. Q-learning is used to compute required arm motions for executing the pivoting.

A dual arm robotic platform is used in [5]. Both arms are equipped with 1-DOF parallel grippers. However, only one of the grippers is used to grasp the object,

whereas the other gripper is used as an external surface that enables pivoting and prehensile pushing primitives.

Holladay et al. proposes a general framework for pivoting in [15], including multiple formulations based on the external effects used for pivoting. The first formulation is for pivoting through contact with external surfaces and gravitational effects, using quasi-static models, whereas the second one utilizes inertial effects, and is modeled using dynamic formulations. In [50], a pinch grasp with the object is modeled as a revolute joint with friction. Authors develop a sliding mode controller that controlles the finger opening in a parallel jaw gripper to modulate the joint friction. Pivoting is dynamically modeled accounting for the inertial and gravitational effects. Shi et al. utilizes a two fingered gripper setup in horizontal plane to perform in-hand sliding via inertial effects using dynamic hand motions in [6]. Their dynamic formulation of sliding is based on a soft-finger limit surface model. Stepputtis et al. uses a learning based method in [51] to predict slipping conditions from tactile data to perform in-hand pivoting through gravity and inertial effects. In [52] a novel sliding motion model is developed and two algorithms based on the sliding model are proposed. The first algorithm aims to reduce the gripping force of the parallel gripper, while avoiding object slip. The second algorithm controls rotational sliding to perform in-hand pivoting. Their approach relies on force/torque sensors and object orientation measurements.

Although extrinsic dexterity enables a wide range of object motion for grippers that otherwise cannot be achieved, the motion is still limited to a plane along which objects can only pivot or slide. To perform extrinsic dexterity based manipulation, either external surfaces or robotic arm motions are required. As frequently mentioned in this section, dynamic formulations of contact and inertia are usually required to model and execute these manipulation primitives. Existing works in the

literature mostly assume known dynamic models of objects and external surfaces. These requirements and limitations may pose challenges when trying to achieve a higher DOF manipulation on everyday objects in unknown environments.

## 2.3 In-Hand Manipulation Planning and Control

In Sections 2.1 and 2.2, mechanisms that are intrinsically capable of in-hand manipulation and techniques that enable in-hand manipulation are discussed. For some of these works, we mentioned the corresponding control, planning, or learning-based approaches as well. Now, we provide a detailed discussion about the planning, control, and learning-based methods for in-hand manipulation.

Graph-based methods are widely used to perform in-hand manipulation planning. Trinkle and Hunter acknowledge the large state space involved in the manipulation planning problem and decompose it using the contact configurations in [1]. They derive the contact mechanics for a multifingered system assuming sliding contacts can occur and determine joint trajectories by searching the decomposed graph. In [53], a two stage planning algorithm is developed, that determines subgoals within a configuration space graph, and seekes quasi-static trajectories to achieve the subgoals. They use a multifingered hand assuming no finger-gaiting but contact mode switches between rolling, sliding,and twisting. A random search is utilized on the graph to find trajectories. Another work that uses multifingered hands is [54], in which Saut et al. considers a grasp space, which is divided into subspaces depending on the finger configurations. Fingers are modeled to have sharp contacts with the object that prevents rolling effects. However, they can be relocated on the object to change the finger configuration, enabling transitions between different grasp subspaces. Proposed approach aims to explore the grasp space to find finger trajectories

and relocations to follow desired object trajectory.

Graph based methods are also used along with underactuated, adaptive hands and extrinsic dexterity based approaches. Hou et al. uses a two fingered gripper to achieve a pinch grasp in [55], which can be adjusted via on table pivoting and rolling primitives. A quasi-static formulation is adopted that neglects inertial effects. Assuming no slip between the hand and the object and coulomb friction between the object and the table, proposed planner aims to find object and hand orientation trajectories and a cartesian trajectory to execute the required rotations. Utilized graph consist of grasp poses connected through the primitives. They solve for a sequence of pivoting and rolling using Dijkstra's algorithm. Cruciani et al. uses a dual-arm system in [5], where one of the hands is used as an external support for pushing and pivoting actions. They generate a dexterous manipulation graph, whose nodes are connected through the predefined primitives. A path corresponding to a sequence of pushing and pivoting is found using Dijkstra's algorithm. Primitives are formulated and executed in kinematic level. In following work [56], dual-arm system is used for regrasping operations, where the main hand is allowed to release the grasp and regrasp the object while the grasp is transferred to the secondary hand in between. Regrasping operation allows the hands to reach contacts on different faces of the object surface, which previously corresponded to disconnected components proposed in the dexterous manipulation graph. Cruciani et al. propose a learning-based method to learn shape priors and utilizes them to estimate object shapes under partial visual observation in [57]. Using this approach dexterous manipulation graphs can be incrementally built for unknown objects.

Bircher et al. use an underactuated hand in [58] and propose an energy gradient-based method for planning in-hand manipulation. This approach does not require any contact or dynamics modeling, which previously posed a challenge for underac-

tuated hands. By calculating energy states of different system configurations under a given input, an energy map is generated. The gradient of this map is used to determine the object trajectory. Discretizing this map leads to a connected graph, on which a path can be found to reach desired goal configurations using breadth-first search. Calli et al. use the adaptive hand Yale Model T-42 in [59] and propose a planning algortihm based on vision-based control and estimation of grasp safety. They use a heuristic guided planner iSST, which utilizes Dynamic Time Warping as a measure of curve similarity to compute the node costs. Using visual data-driven regression process, based on the previous work [60], which learns to predict different modes of in-hand manipulation, authors can estimate the probability of dropping the object during manipulation. The planner is combined with the estimation method to avoid near failure states.

Trajectory optimization is another approach used to plan for in-hand manipulation. In [61], Han et al. propose a general technique that considers contact kinematics, nonholonomic motions, and grasp stability to generate and track joint trajectories for a hand with two flat fingers. The goal is to track a reference trajectory with the object while optimizing grasp quality. In [9], trajectory optimization is used to synthesize hand motions, given high-level goals such as desired object pose. Resulting plan specifies the location and time for the finger contacts and their trajectory. Sundaralingam and Hermans propose a method based on kinematic trajectory optimization in [7]. They assume no finger-gaiting, only allowing the reconfiguration of the object within hand. However, instead of formulating finger contacts as rigid constraints, they define them using cost functions, resulting in a relaxed constraint, which only penalizes the sliding and rotation of the fingers. The advantage of this method is the kinematics level modeling, and a joint space solution that does not require inverse kinematics solvers. In [62], they extend the work by incorporating

trajectory smoothing, collision avoidance, and object pose feedback. Developed in-grasp manipulation strategy is combined with a finger-gaiting approach in [63]. The proposed planner solved two different optimization problems to alternate between in-grasp manipulation and finger-gaiting to achieve desired grasps.

Daoud et al. propose a two phase approach in [64] to plan the finger motions and fingertip forces on an anthropomorphic hand. They use geometric models for contact to solve for finger motion, whereas a neural network based force evaluation method is used to estimate forces without sensing and an optimization algorithm is utilized to control the finger tip forces. This approach achieves desired object motion, while maintaining the grasp stability through force control. An online planning algortihm is proposed in [65], that considers a multifingered hand and arm system. By analyzing the workspace of the hand, the planner assigns weights for the contributions of the hand and the arm to coordinated motion. Their method relies on pose error computation and cartesian position or impedance controllers for execution. A sampling based planning approach is used in [66]. This work considers extrinsic dexterity based manipulation relying on external surfaces for pushing and pivoting of objects. Chavan-Dafle and Rodriguez propose a low-level optimization for the inverse dynamics of the manipulation primitives and a sampling-based high-level planner using transition-based RRT* (T-RRT*) to find a sequence of manipulation primitives. They assume known mechanical properties of the object such as mass, geometry, inertia, and friction coefficients and the gripping force. In [49], authors replace the low-level planning component, using the proposed motion cones model.

To finalize the review of in-hand manipulation planning and control, we mention the learning-based methods. Learning methods are most commonly utilized to control the motion of multifingered and high-DOF hands. Most learning-based methods

aim to learn actuation inputs or controllers for in-hand manipulation. In [14], the TWENDY-ONE Hand is used. The hand is equipped with force/torque sensors at fingertips and tactile arrays on the skin. Proposed method uses a feedforward neural network to learn the next motor positions for a predetermined manipulation motion using the data on current grasp and object geometry. In [67], Kumar et al. use a pneumatically actuated hand capable of in-grasp manipulation and finger-gaiting. Using model-based reinforcement learning, a time-varying linear Gaussian controller is learned, that is directly used to compute the required pneumatic system inputs without the need for any additional mapping in joint space. Cost functions used in the learning scheme is based on the reference object poses. In [68], model-free deep reinforcement learning is used to learn in-hand manipulation policies (joint trajectories) for motions such as object relocation, in-hand repositioning, door opening, and tool use. Authors investigate two approaches, learning from scratch vs. learning through human demonstrations. OpenAI uses the Shadow Hand in [8] and perform reinforcement learning to learn a controller policy that computes the required actuator positions for a vision-based object reorientation within-hand. In contrast to other approaches that learn actuation policies, deep dynamic models are learned in [69]. Authors use the learned models with model predictive control (MPC) to perform in-hand reorientation, writing, and ball juggling.

There are also learning-based approaches that use compliant and adaptive hands. In [70] a three-finger compliant hand equipped with tactile sensors is used to learn a control policy. They formulate the in-hand manipulation as a Markov Decision Process (MDP), where the state parameters are the finger angles and tactile sensor measurements. Actions correspond to the finger velocity inputs. Proposed reward function for the reinforcement learning is based on position and force error. Li et al. use a 3 fingered hand on a 2D plane in [71] that is capable of reposing,

sliding, and flipping slender objects. They propose a low-level planner through dynamic formulations to execute the primitives using torque controllers and a high-level controller based on deep reinforcement learning to decide for a sequence of manipulation primitives and corresponding parameters. Their method assumes that inertial properties of the object are known. In [72], Sintov et al. use the adaptive hand Yale Model T-42. They propose a learning-based planning scheme that require the learning of a transition model and a model critic, where the critic is responsible for estimating the model error. Authors train neural networks for both components and solve a Non-Observable MDP problem in an open-loop manner using the learned model and critic. Proposed approach aims to learn an optimal control sequence that optimizes the cost function, reaches the goal state, and avoids the regions that are inaccurately modeled which are identified by the critic.

# Chapter 3

# 3D In-Hand Manipulation

In this work, we consider an in-hand manipulation task that requires the contact between the hand and the object to be navigated from an initial region to a goal region on the object. As shown in Figure 3.4a, previously used setup for the variable friction hand was restricted to 2D in-hand manipulation, where the object was able to translate and rotate within-hand along a plane (represented by green arrows). Using these manipulation primitives, only a small set of goal regions were reachable. Here, we expand the workspace to 3D to include the degrees of freedom shown in Figure 3.4b by extrinsic dexterity-based manipulation strategies (represented by yellow arrows). With the improved action space, we can reach a larger set of goal regions. This chapter introduces the variable friction hand design and the manipulation primitives enabled by the hand setup. Then it describes and models the extrinsic dexterity based manipulation strategies utilized to expand the manipulation workspace to 3D.

## 3.1 Variable Friction Hand

The variable friction hand is an open-source design from Yale OpenHand Project[73], named as Model VF, described in detail in [18]. Model VF consists of two fingers connected to the hand base by rotary joints. As shown in Figure 3.1, each finger is equipped with a friction switching mechanism, consisting of a servo motor, a low friction insert, rubber bands, and tendons. Fingers are by default at low friction state, where the rubber bands keep the low friction surfaces exposed. Fingers can be set to high friction state by actuating motors to pull on the tendons connected to the low friction inserts, retracting them and exposing the high friction surface on the fingers. Dynamixel XM430-W350 motors are used to actuate the finger joints. In this setup, Dynamixels are either used in the position control mode or torque control mode, depending on the manipulation primitive to be executed.

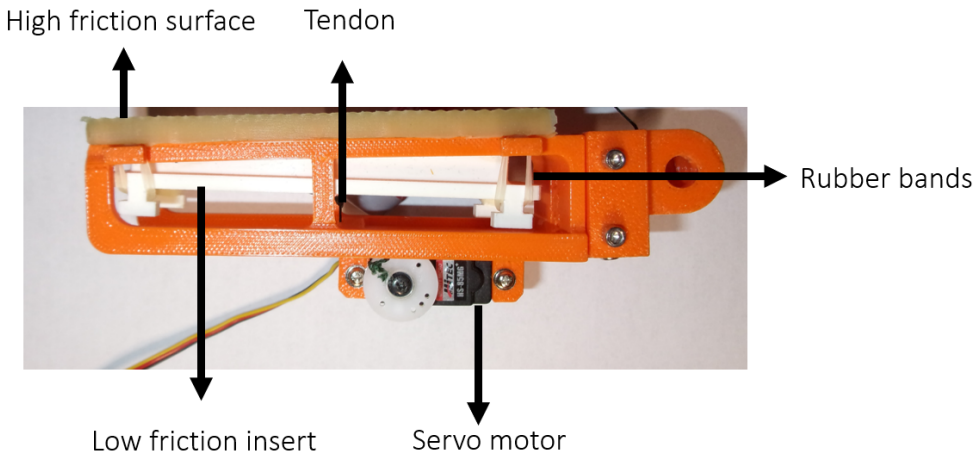


Figure 3.1: Finger equipped with a friction switching mechanism

The friction switching mechanism enables the control of the contact mode between the fingers and the object by imposing sticking or sliding behavior. The main logic of the variable friction finger design is to have the friction coefficients of the high and low friction surfaces at extremes, which is sufficiently achieved by the

materials used in the current fingers. This property enables controlled sliding and rotation of a wide range of objects with different friction characteristics. Following manipulation primitives are enabled solely by the variable friction hand.

### 3.1.1 In-Hand Sliding

To achieve in-hand sliding, one of the fingers is switched to low friction state, whereas the other finger is at high friction state. Slides can be performed in both distal and proximal directions. Four distinct manipulation primitives emerge from these conditions. Table 3.1 summarizes the friction states, Dynamixel control modes, and directions of object and finger motion corresponding to each sliding primitive. Figure 3.2 demonstrates how each sliding primitive is executed using variable friction fingers.

Table 3.1: In-hand sliding primitives

| Primitive | Friction States | Object Direction | Finger Direction | Dynamixel Mode |
|---|---|---|---|---|
| Slide Left Down | (low,high) | Proximal | CCW | (position,torque) |
| Slide Left Up | (low,high) | Distal | CW | (torque,position) |
| Slide Right Down | (high,low) | Proximal | CW | (torque,position) |
| Slide Right Up | (high,low) | Distal | CCW | (position,torque) |

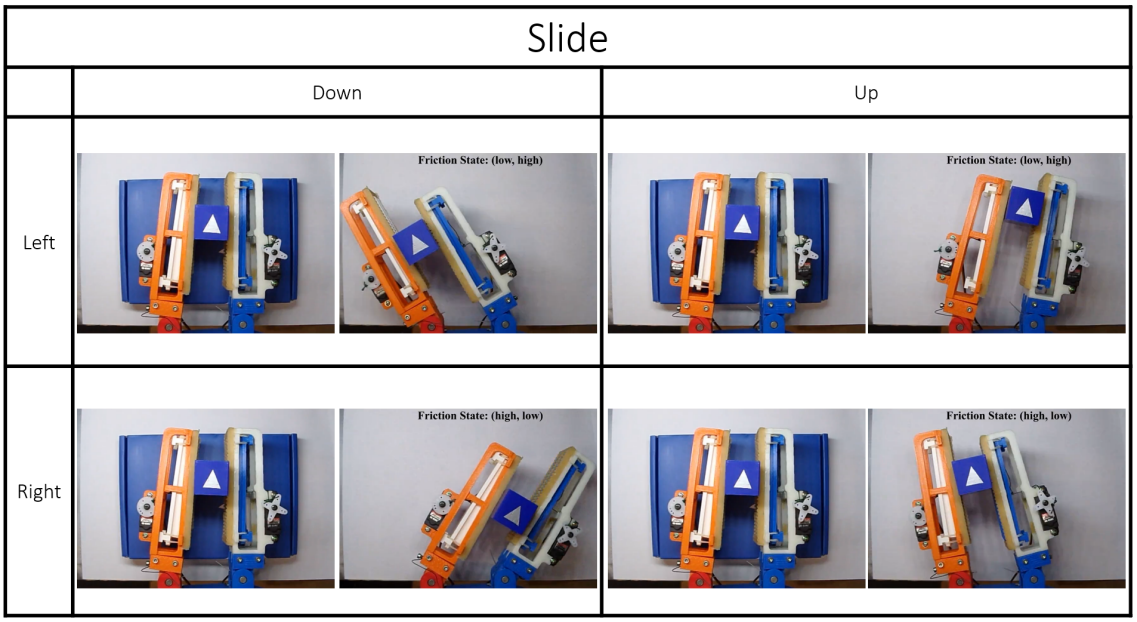


Figure 3.2: In-hand sliding using variable friction fingers.

### 3.1.2 In-Hand Rotation

To achieve in-hand rotation, both fingers are kept at high friction state. Object in grasp pivots around the contact with the fingers, enabling object rotation. Table 3.2 summarizes the friction states, Dynamixel control modes, and directions of object and finger motion corresponding to each rotation primitive. Figure 3.3 demonstrates how each rotation primitive is executed using variable friction fingers.

Table 3.2: In-hand rotation primitives

| Primitive | Friction States | Object Direction | Finger Direction | Dynamixel Mode |
|---|---|---|---|---|
| Rotate Clockwise | (high,high) | CW | CCW | (position,torque) |
| Rotate Counter-clockwise | (high,high) | CCW | CW | (torque,position) |

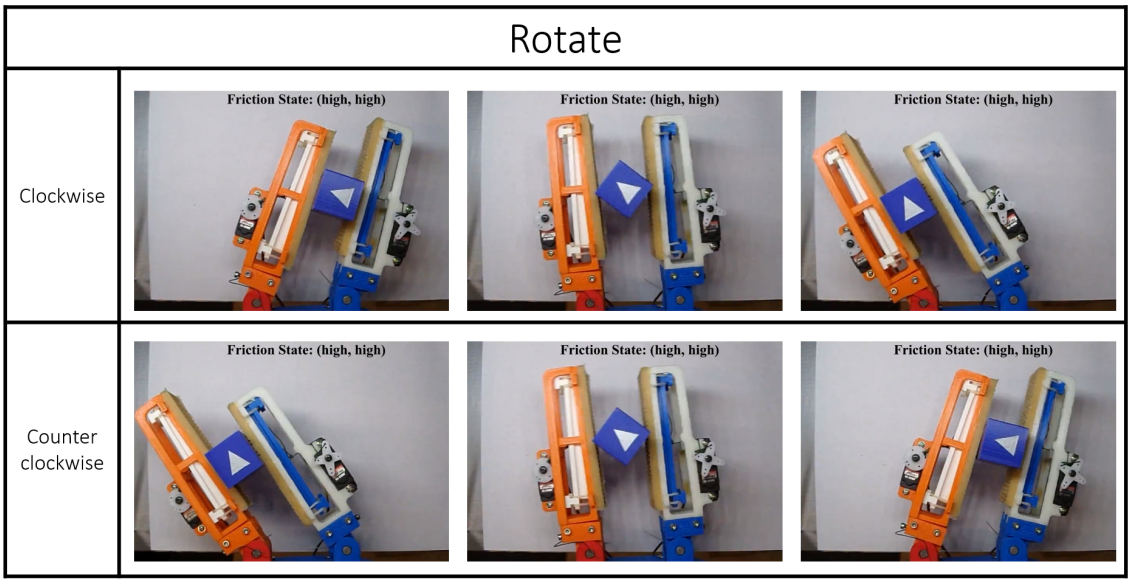


Figure 3.3: In-hand rotation using variable friction fingers.

### 3.1.3 Kinematics Model

As the contact modes vary with the friction states of the fingers, different kinematics models are required for each finger state combination. Forward and inverse kinematic models were derived in [19].

In this work, we use these kinematics models to compute required position inputs to the Dynamixels, when executing the planned sliding and rotation primitives.

## 3.2 Extrinsic Dexterity Based Strategies

By attaching the variable friction hand to a robotic arm, we introduce additional degrees of freedom to the manipulation platform. Following manipulation primitives are performed using extrinsic dexterity-based manipulation strategies, such as prehensile pushing, pivoting and exploitation of gravity. These manipulation strategies are enabled by the variable friction fingers, the motion of the robotic arm, and the horizontal support surface in the environment.

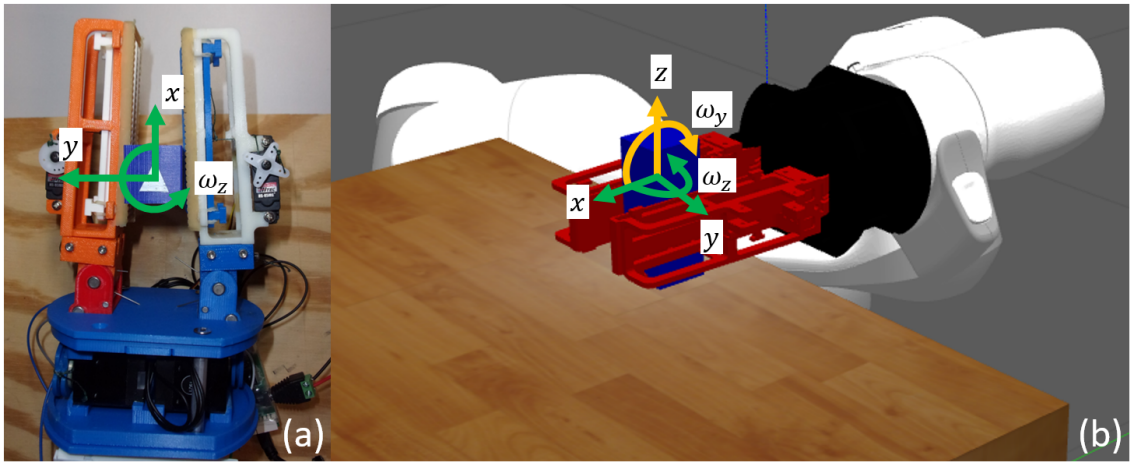


Figure 3.4: (a) In prior work, a 2-DOF robot hand was proposed that is capable of translation and rotation in 2D. (b) By combining this hand with a robotic arm, extrinsic dexterity based manipulation primitives such as prehensile pushing and pivoting can be utilized to achieve 3D manipulation.

### 3.2.1 Moving Contact Up and Down

Contact between the fingers and the object is translated in positive $z$-direction (Figure 3.4b) by exploiting the gravitational forces acting on the object. To achieve this action, both finger frictions are switched to the low friction state and fingers slide up on the object while the contact is maintained with the support surface. Different from the existing work on the extrinsic dexterity based manipulation techniques, this can be achieved via a simple kinematic model, without using any force sensing or control, thanks to the friction modulation. By controlling the end-effector position of the robotic arm along the $z$-axis, required sliding can be performed to move the contact up by the planned amount.

Likewise, having the contact friction low, contact between the fingers and the object is translated in negative $z$-direction through prehensile pushes against the supporting surface. Same contact assumptions and kinematic model hold for controlling the amount of pushing. After these operations the friction states are switched to high to recover the firm grip.

### 3.2.2 Pivoting

Similarly, being able to change finger surface friction allows us to leverage extrinsic contacts reliably without the need of force control or dynamic models. A kinematic model is derived by modelling contacts between the fingers and the object and between the object and the supporting surface as frictionless revolute joints. Frame assignments are shown in Figure 3.5. Using the Denavit-Hartenberg parameters provided in Table 3.3, transformations between the end-effector frame and the object frame is found. Similar to the stage-wise pivoting in [11], pivoting is executed according to the kinematic model in two stages as shown in Figure 3.6:

1. Maintaining the contact angle with the fingers at high friction state, object is rotated around the pivoting axis by controlling the end-effector pose via robot motions.

2. Keeping the high friction state, the robot moves the object to establish a contact with the support surface at the pivoting point. The hand switches to low friction state for both fingers, and the robot follows an arch motion based on the derived kinematics.

Table 3.3: DH Parameters for arm-hand system

| Transformation | $\theta$ | $d$ | $a$ | $\alpha$ |
|---|---|---|---|---|
| $T_h^{ee}$ | $\frac{3\pi}{4}$ | $d_1$ | 0 | $\frac{\pi}{2}$ |
| $T_f^h$ | $\theta_{finger}$ | 0 | $d_2$ | $\frac{\pi}{2}$ |
| $T_d^f$ | $\theta_{contact} - \frac{\pi}{2}$ | 0 | $d_3$ | 0 |
| $T_p^d$ | $-\frac{\pi}{2}$ | 0 | $d_4$ | $\pi$ |
| $T_o^p$ | $\theta_{pivot}$ | 0 | 0 | 0 |

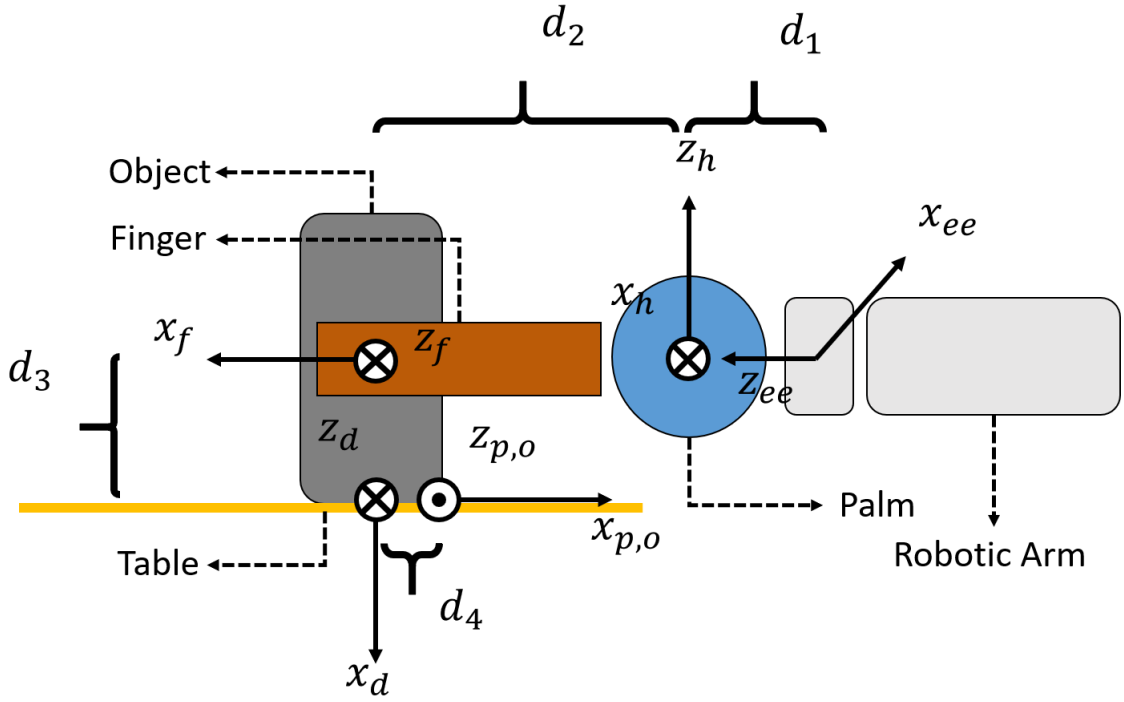


Figure 3.5: Frames are attached on the robot end-effector, gripper palm, finger contact, object and pivoting center. Parameter $d_1$ is a fixed offset depending on the arm-gripper setup. Planning algorithm outputs the current finger parameters ($\theta_{finger}$, $d_2$, $d_3$ and $d_4$) for a pivoting action, which are then used in the kinematic model. During the pivoting, finger parameters are held constant, while the angles of the virtual joints ($\theta_{contact}$ and $\theta_{pivot}$) vary.

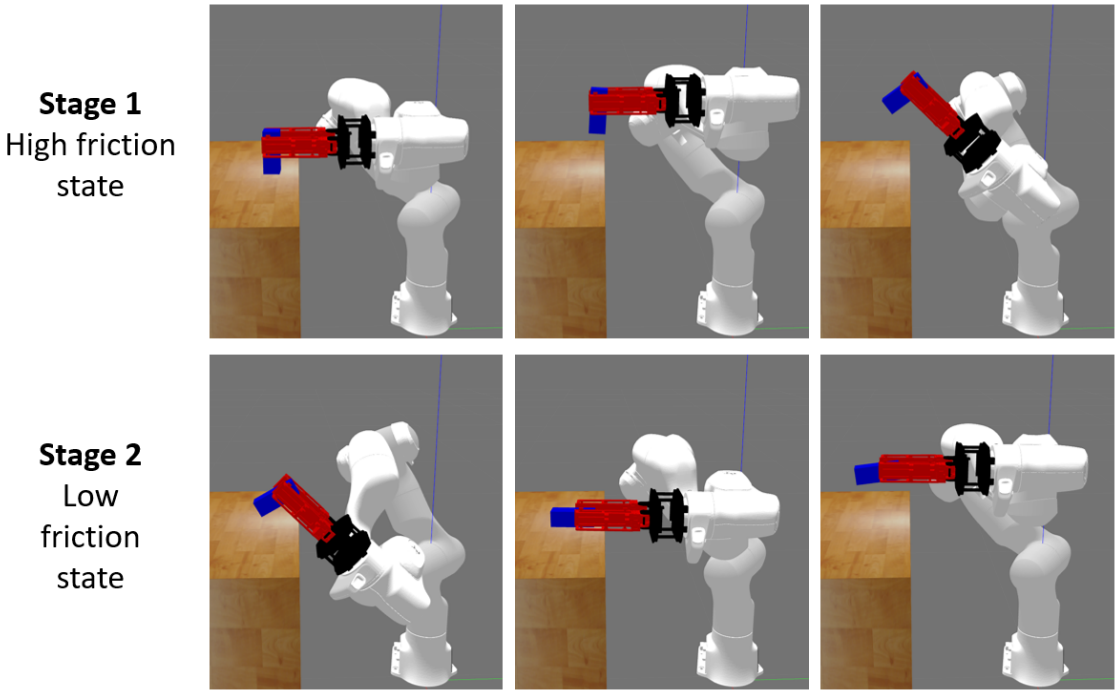


Figure 3.6: Pivoting is executed in two stages. At Stage 1, fingers are in high friction state, as the object rotates around the pivoting axis. At Stage 2, object is simultaneously rotated around the pivoting axis and finger contact by utilizing the table surface and switching to low friction state.

# Chapter 4

# Region-Based Manipulation Planning

Considering an in-hand manipulation task that requires the contact between the hand and the object to be navigated to a desired region on the object, we develop a novel region-based motion planning formulation, based on a technique to measure the distance between convex polygonal patches. This chapter introduces the region-based planning problem and describes the proposed A* search algorithm to solve for manipulation sequences.

## 4.1 Problem Formulation

Region-based within-hand manipulation planning problem is defined as finding a sequence of manipulation primitives $\pi = [a_1, ..., a_T]$ that navigates the contact between the hand and the object from an initial state $s_0$ (the initial contact region) to a set of desired regions $P = \{p_1, ..., p_n\}$ at terminal step $T$. Regions $p_1$ to $p_n$ are on different surfaces of the object through $1 - n$. For the variable friction hand, there

is a contact region between each finger and the object to be navigated within one of the goal regions. In this work, we have the following assumptions:

1. 3D geometry of the object is assumed to be known and provided as a list of vertices.

2. Objects are prismatic, object faces and contact regions are convex polygons.

3. Variable friction fingers make contact with two parallel surfaces on the object with opposing normals.

Our motion planning algorithm aims to solve the optimization problem given as:

$$\min_{\pi}(E(s_T, P) + w \sum_{t=0}^{T-1} g(a_t)) \tag{4.1}$$

where $E(s_T, P)$ is a cost function designed to minimize the contact region that is left outside of the goal region at terminal step $T$, $g(a_t)$ is the cost of taking an action $a$ at time $t$ and $w$ is a trade-off weight between the cost components. Depending on the limitations in state space, kinematic model and hardware, constraints can be imposed on the proposed optimization problem.

## 4.2 A* Search Algorithm

We solve the optimization problem defined in Section 4.1 with a modified A* algorithm and a region-based heuristic design. Different from the motion planning approaches in the literature that consider a point-based agent being navigated into a goal pose, a goal state, or a set of goals, we consider a polygonal agent that will be navigated inside a goal region.

### 4.2.1 States

For the region-based planning problem, the states of the system are the contact regions between the fingers and the object represented by convex polygons in 2D. A state $s(f_l, f_r, C_l, C_r)$ stores the following information:

- Indices of the object faces (through $1-n$) in contact with left and right fingers: $(f_l, f_r)$
- Contact regions for left and right fingers represented using the vertices of contact polygons: $(C_l, C_r)$

### 4.2.2 Actions

Actions available for this system are the manipulation primitives described in Chapter 3. There are 6 manipulation primitives performed by the variable friction hand. 3 additional manipulation primitives are enabled by the cooperation of the robotic arm and the hand. Total of 9 actions are considered in the motion planning problem, resulting in the following action space: $A = \{$Slide along the left finger up, slide along the left finger down, slide along the right finger up, slide along the right finger down, rotate clockwise, rotate counterclockwise, move contact up (along the z direction in Figure 3.4b), move contact down, pivot$\}$.

Each of these actions are discretized as follows. The resolution of slides and moving contact up-down are dependent upon the range of dimensions of the objects being manipulated. The resolution is expected to allow multiple slides along the object surface. However, having a small resolution creates a burden for the planning algorithm by expanding the state space and increases the planning time of the required manipulation sequence. Based on these factors, the resolution is determined experimentally for a group of test objects. The resolution of in-hand rotation and

pivoting actions depend on the outer angles of the polygonal object faces, which are automatically detected by the planning algorithm. Figure 4.1 illustrates the rotation and pivoting resolutions on a hexagonal prism.

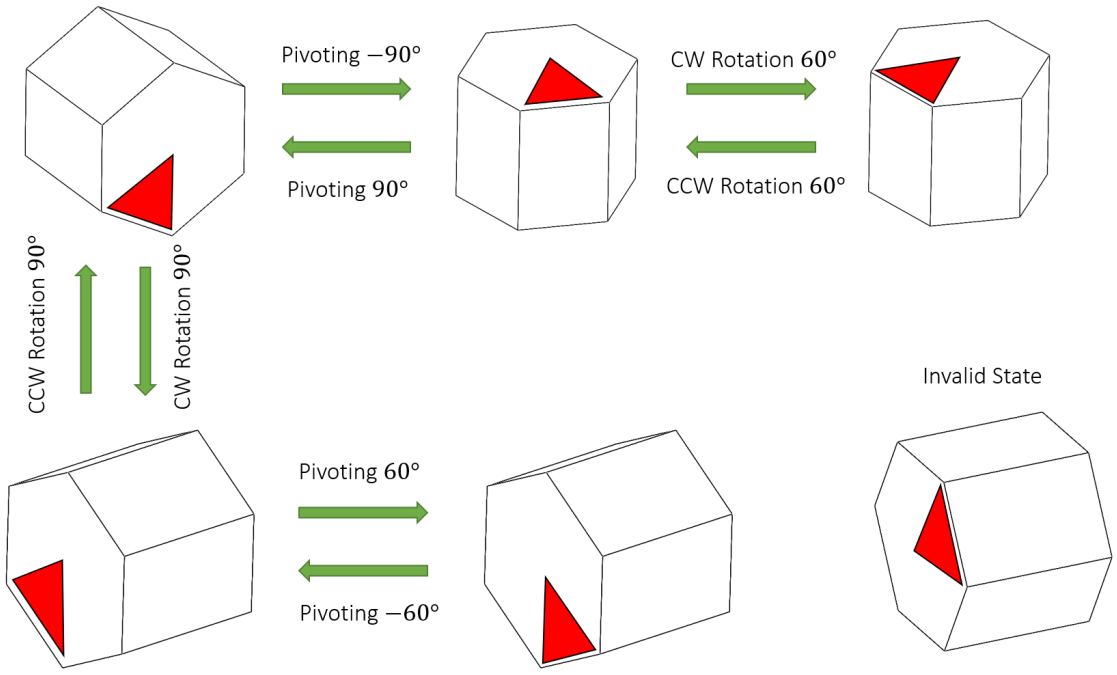


Figure 4.1: Pivoting and rotation resolutions for a hexagonal prism. The invalid orientation results in edge contacts when rotated, hence it is an invalid state.

### 4.2.3 Transition Model

Our transition model illustrated in Figure 4.2 describes how taking an action $a$ transforms the current state $s$ to the next state $s'$ (e.g. how pivoting the object changes the contact region between the fingers and the object). For any given object and a set of goal regions complying with our assumptions, proposed algorithm can automatically determine the valid manipulation primitives at a current state and use the transition model to transform a it into to the next state.

Table 4.1 summarizes the effect of each primitive on the system state $s$. Fig-

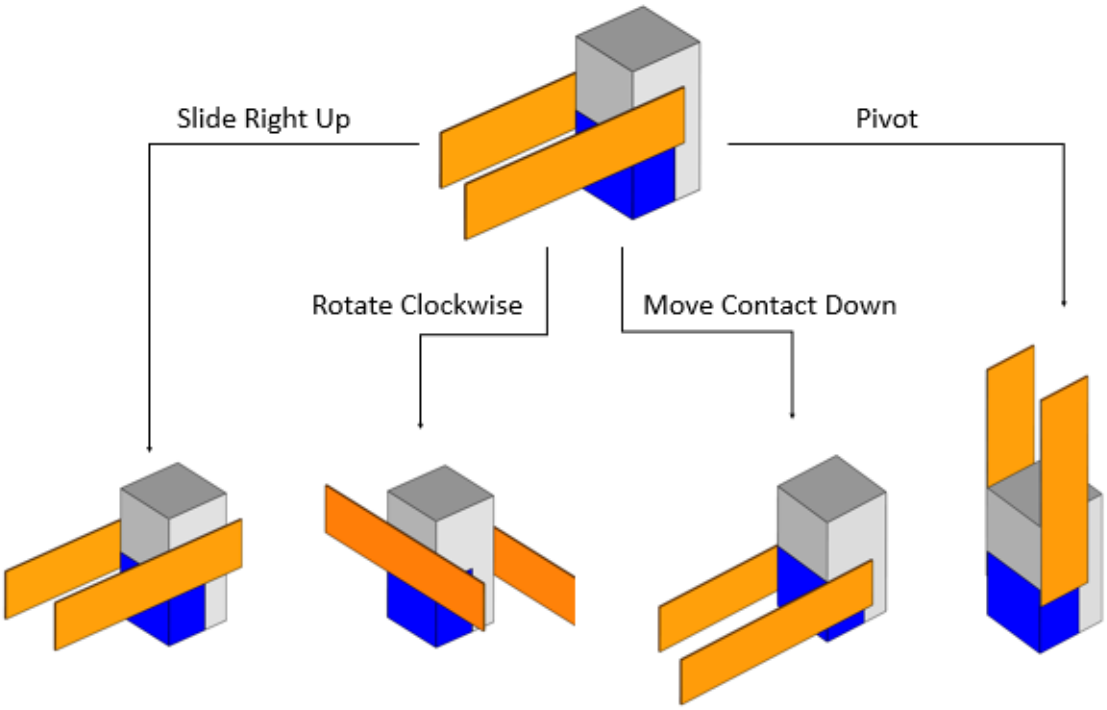


Figure 4.2: When an action is taken, contact regions are transformed according to a transition model. Following an action, the contact region can translate and rotate on the same surface or travel between different surfaces.

ure 4.3 is an accompanying illustration for a square prism. Note that contact faces are only changed as a result of in-hand rotation actions. The remaining primitives only move the contact patches along the same surface. Sliding actions have the capability to increase and decrease the contact area depending on the object position. For states in which the object is entirely within the palm, slides cannot change contact area. For states in which the object is at the finger tips, slides can effect the contact area. Rotation actions generate a similar effect. Moving contact up and down locates the contact patches vertically along object surface, whereas pivoting rotates the contact patches around the pivoting center.

Table 4.1: Effects of each primitive on system state parameters. NE denotes no effect.

| Primitive | $f_l$ | $f_r$ | $C_l$ | $C_r$ |
|---|---|---|---|---|
| Slide Left Down | NE | NE | NE/increase in area | NE |
| Slide Left Up | NE | NE | NE/decrease in area | NE |
| Slide Right Down | NE | NE | NE | NE/increase in area |
| Slide Right Up | NE | NE | NE | NE/decrease in area |
| Rotate Clockwise | contact face change | contact face change | NE/increase in area | NE/decrease in area |
| Rotate Counter-clockwise | contact face change | contact face change | NE/decrease in area | NE/increase in area |
| Move Contact Down | NE | NE | moves down | moves down |
| Move Contact Up | NE | NE | moves up | moves up |
| Pivot | NE | NE | rotates around pivoting center | rotates around pivoting center |

### 4.2.4 Termination

The A* search algorithm terminates with success, when a sequence of manipulation primitives, $\pi = [a_1, ..., a_T]$, is found that navigates the contact regions for both left

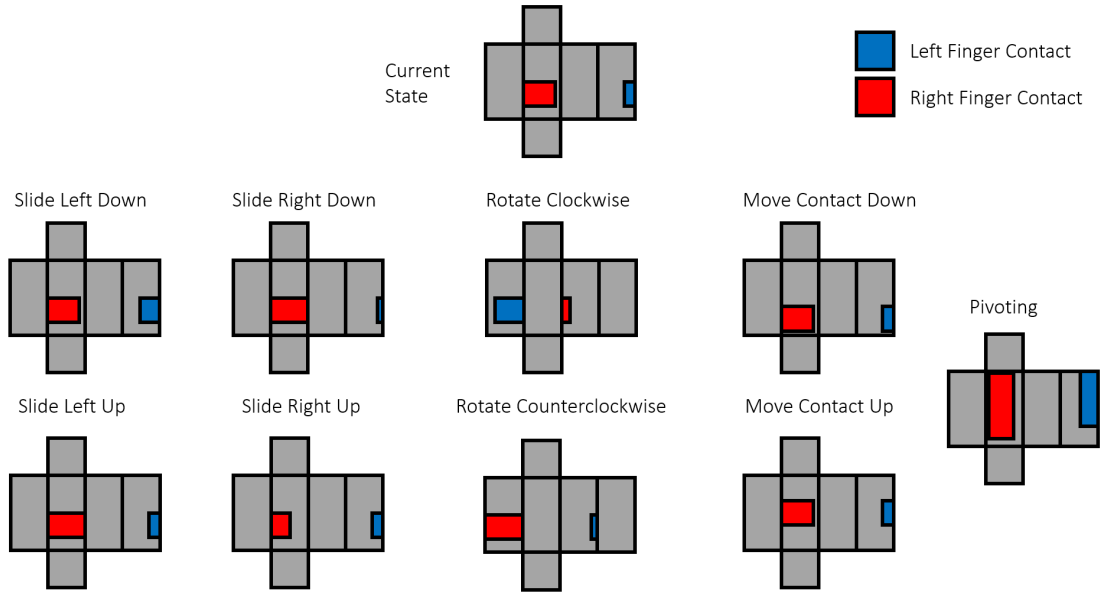


Figure 4.3: This figure illustrates the changes in contact regions for left and right fingers for each motion primitive on a square prism.

and right fingers into goal regions on the object surface, $P = \{p_1, ..., p_n\}$. In other words, it terminates successfully, when the path $[s_0, a_1, s_1, a_2, ..., s_T]$ is generated for $s_T(f_l, f_r, C_l, C_r)$ with $C_l \in p_{f_l}, C_r \in p_{f_r}$. The algorithm fails if no paths leading to a termination state $s_T$ is found after exploring the reachable nodes.

### 4.2.5 Cost Function

During the A* search, a cost is assigned to each node/state, consisting of two components. The first component $g(s)$, corresponds to the cost accumulated to reach the current state, whereas the second component $h(s)$, is the cost that will be accumulated to reach the goal state starting from the current state [74]. The total cost associated with a state is found as:

$$f(s) = g(s) + h(s) \tag{4.2}$$

A* search algorithm tries to find the path between initial and goal states with the cheapest cost and it is optimal for heuristic function, $h(s)$, satisfying certain characteristics [74].

### Path (Action) Cost

While determining the costs of manipulation primitives, we consider the in-hand sliding as the reference case, where the cost of a unit sliding is equal to the sliding resolution. In-hand rotation is a slightly complex primitive that requires a larger finger motion compared to unit sliding. Moving the contact up or down requires motion of the arm and pivoting primitive requires even larger arm motion. As this work aims to find a manipulation sequence with least complexity and shortest execution time, we scale up the action costs of these primitives according to Table 4.2.

Table 4.2: Action cost multipliers for each primitive category.

| Primitive category | Cost multiplier |
|---|---|
| Slides | 1 |
| Rotations | 3 |
| Move Up & Down | 10 |
| Pivot | 25 |

### Heuristic Cost

The heuristic function is used to attract the search for the manipulation sequence towards the goal regions. Whereas it is common to use some well-known distance measures such as Manhattan distance or Euclidean distance in a conventional navigation scenario on a 2D grid, this novel problem requires a measure of distance

between two polygonal areas, namely the contact regions and the goal regions on the object surface. Since manipulation primitives move the contact region along the surface of the object, distances to be measured also needs to be along the object surface. Geodesic distance computation methods are developed for this purpose [75]. However, a quicker and simpler approach is required to be implemented in the proposed planning algorithm. We propose a method that automatically unfolds the surfaces of a given object and projects them onto a plane. This approach ensures that the euclidean distances measured on the projection plane are the same with the distances that are measured along the object surface. Figure 4.4 illustrates the unfolding procedure and the computation of the heuristic.

Let $c_i$ denote the $i$-th vertex of the contact region between the fingers and the object, and $p_m$ denotes the goal region on surface $m$. The shortest distance between a point and the surface is computed with function $d$. Following steps summarize the details of the heuristic computation:

1. For each corner on a contact region ($c_i$), the shortest distance between the corner and the goal region on a selected surface is computed ($d(c_i, p_m)$). Distances are summed up for 4 corners.

$$D_m = \sum_{i=1}^{4} d(c_i, p_m) \tag{4.3}$$

2. Step 1 is repeated for each goal region, $P = \{p_1, ..., p_n\}$.

3. Minimum of the stored sums is the heuristic at state $s$ for the selected contact region.

$$h(s) = \min(D_1, D_2, ..., D_n) \tag{4.4}$$

4. Heuristic is computed for both right and left contact regions. Final heuristic

is the sum of the right and left heuristics.

$$h_{final}(s) = h_l(s) + h_r(s) \tag{4.5}$$

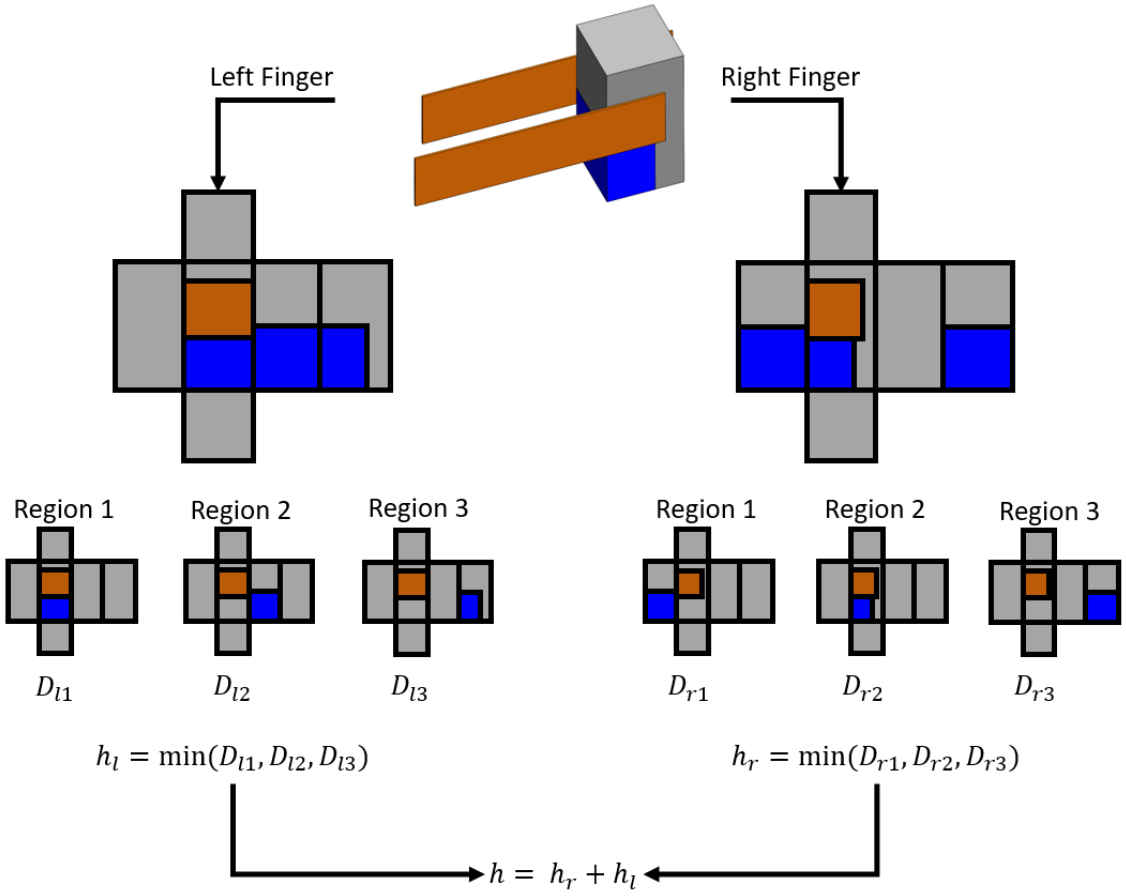


Figure 4.4: To compute the heuristic function, surfaces of the object are unfolded, while keeping the contact surface at the origin. For each goal region the shortest distances with the contact are computed. Minimum of the shortest distances is the heuristic for the selected finger.

As discussed in [19], we modify the heuristic cost using a discount multiplier to favor repetitive actions, according to following principle:

$$h(s_t) = \beta h(s_t) \tag{4.6}$$

where

$$\beta = \begin{cases} 0 < \beta < 1, & \text{if } a_t = a_{t-1} \\ 1, & \text{otherwise} \end{cases} \tag{4.7}$$

This leads to smoother paths in execution, decreasing the number of switches in friction states and Dynamixel control modes, thus the tear and wear of the variable friction hand setup. A similar approach is adopted in [76], to reduce the number of contact switch-overs between extrinsic dexterity based manipulation actions.

Similarly, we adapt the weighted A* search approach, given by Equation 4.8, from [19] to speed up the search, as the state space of the 3D system is even larger. However, this approach is known to sacrifice the optimality by a degree of $\epsilon$ of the A* search.

$$f(s) = g(s) + \epsilon h(s), \quad \text{with } \epsilon > 1 \tag{4.8}$$

The heuristic cost and the assigned action costs do not exactly represent the same measures along the manipulation path, as the heuristic cost is a measure of distance and the action costs are assigned according to action complexity and execution times. This might lead to some concerns regarding the admissibility of the heuristic, thus the optimality of the algorithm. To address this issue, we point out to the definition of the sliding resolution and assigned sliding cost. Assuming that sliding actions enable the fingers to traverse along the object surface both horizontally and vertically as well as across different object faces, the actual cost to any goal region would be equal to the Manhattan distance to that region. The heuristic cost is in the form of a euclidean distance, hence it would be providing an optimistic estimate of the actual cost. Since the costs of remaining primitives are scaled up based on the sliding primitive, the distance measure used as the heuristic is expected to be an optimistic estimate of the required cost, hence an admissible

one in this case.

# Chapter 5

# Experiments and Results

## 5.1 Simulated Experiments

As a proof of concept for the region-based planning approach, we developed a simulation environment. Using the models for the variable friction hand and the Franka Emika Panda arm, the 3D in-hand manipulation platform is spawned in Gazebo [77] on ROS Kinetic distribution [78], as provided in Figure 5.1. The manipulation environment contains a support surface and several prismatic objects. Manipulation sequences produced by the region-based planning algorithm are executed using the simulation environment, confirming our planning and execution approaches qualitatively.

### 5.1.1 Variable Friction Plugin

As it was not feasible to model the actual friction switching mechanism using the low friction inserts that move back and forth between finger gaps, a Gazebo plugin is developed to enable the active control of the friction coefficients during simula-

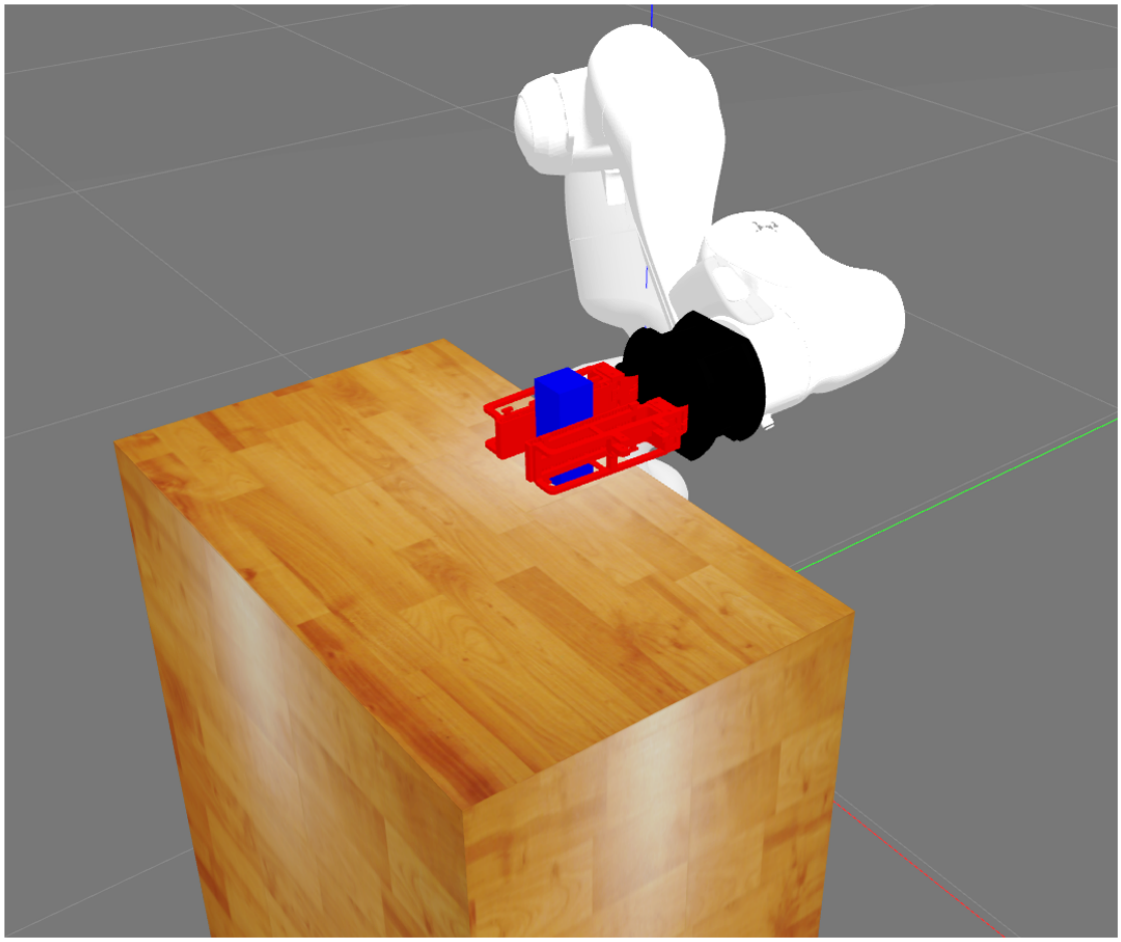

Figure 5.1: Simulation environment for 3D in-hand manipulation platform including variable friction gripper and Franka Emika Panda arm.

tion. The plugin is available in a public repository[1] along with a 2D simulation environment and control algorithms for the variable friction hand.

### 5.1.2 Experimental Framework for Simulation Parameters

Modeling and simulating contact-rich environments are not easy, especially in the presence of continuous impacts and friction. To achieve more stable contact behavior, an experimental framework is developed that enables control over various surface

[1] https://github.com/asahin1/wihm-variable-friction

characteristics during simulation. Using this framework, we have experimented with different sets of simulation parameters in Gazebo, and determined the most suitable one for a stable simulation of the variable friction hand setup. A screenshot of the user interface is provided in Figure 5.2.

### 5.1.3 Simulation Outcomes

Although the developed simulation does not represent the physical system with full accuracy due to the limited capability of modeling the frictional effects, it is a useful tool in assessing the execution and planning of the current and future in-hand manipulation strategies. Furthermore, the simulation environment can be used to generate visual and numeric data to train learning-based algorithms.

## 5.2 Real-Robot Experiments

### 5.2.1 Experimental Setup

To further validate our approach and evaluate the manipulation sequences generated by the planning algorithm, we conduct real-robot experiments using the variable friction hand setup and a Franka Emika Panda arm shown in Figure 5.3. To be used in the experiments, we designed and manufactured the object set given in Figure 5.4 from polylactic acid (PLA) plastic by 3D printing. For each object, 1-3 initial and goal regions are defined. Some examples are shown in Figure 5.5.

We use the in-hand manipulation benchmarking protocol proposed in [79]. 1-3 manipulation sequences are generated for each object corresponding to each initial and goal region. 5 trials are run for each generated plan, with a total of 5-15 executions per object. Average motion planning and execution times are provided

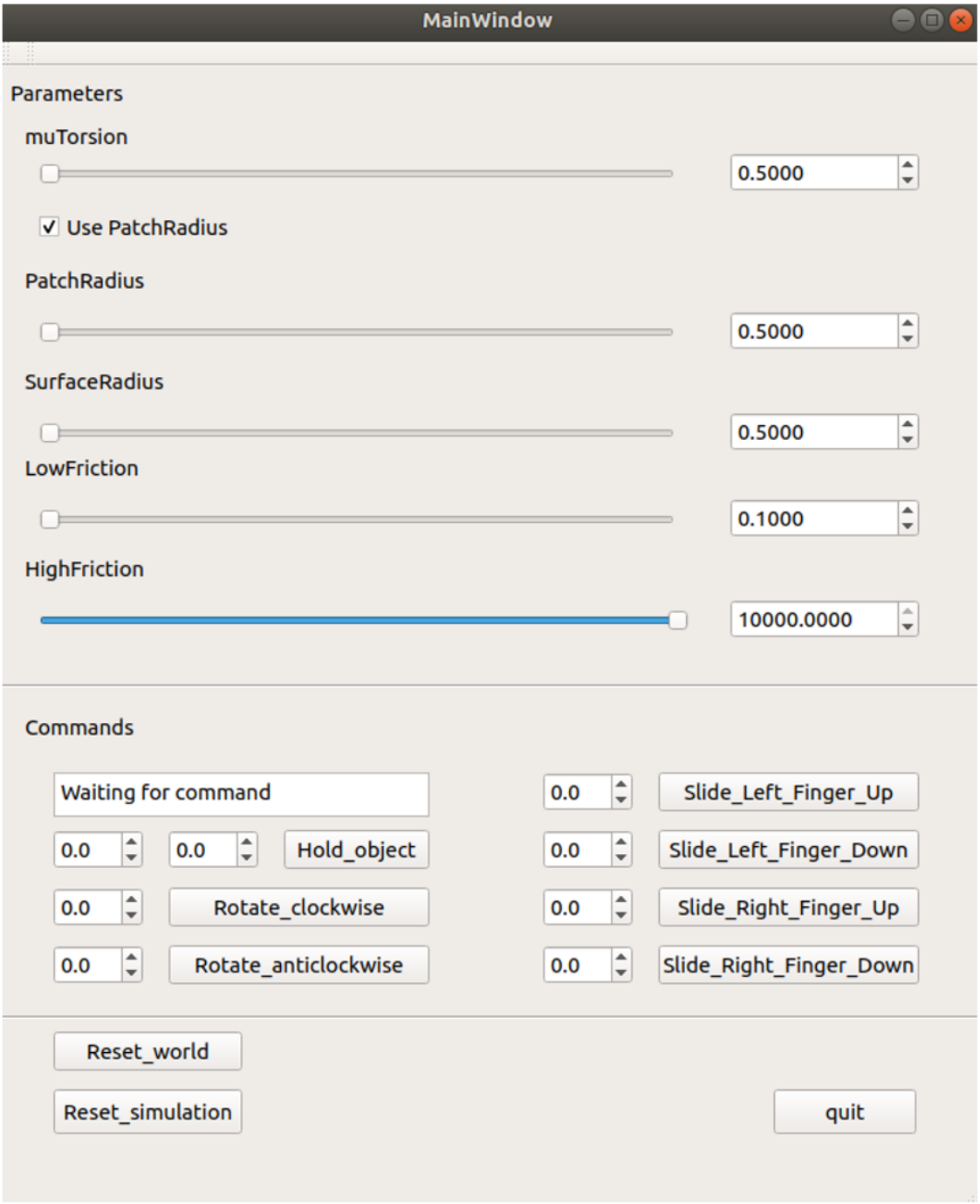


Figure 5.2: User interface for runtime simulation parameter adjustments and high-level in-hand manipulation commands.

along with the failure rates for each object in Table 5.1.

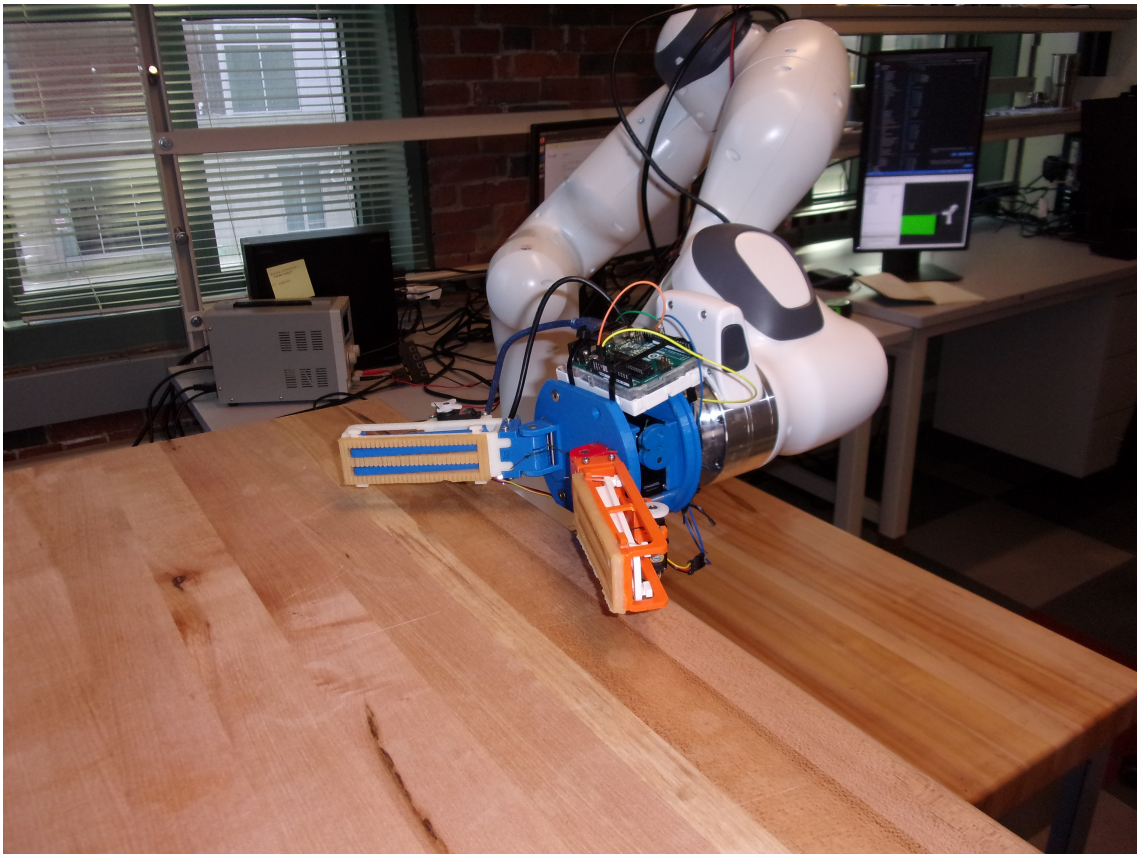

Figure 5.3: Real robot experimental setup including the variable friction gripper and Franka Emika Panda arm.

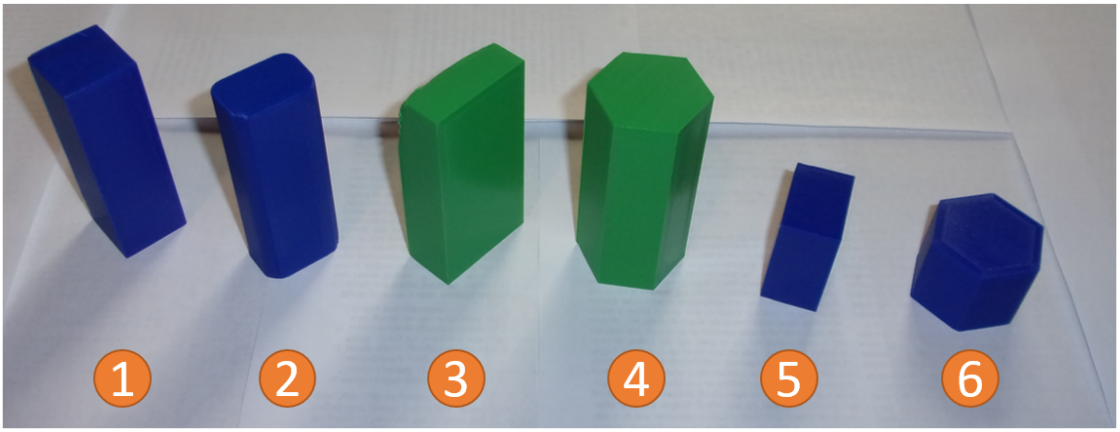


Figure 5.4: Artificial objects are modeled and 3D-printed for the experiments. Set of objects include: 1) square prism 2) rectangular prism curved 3) rectangular prism large 4) hexagonal prism tall 5) rectangular prism small 6) hexagonal prism short.

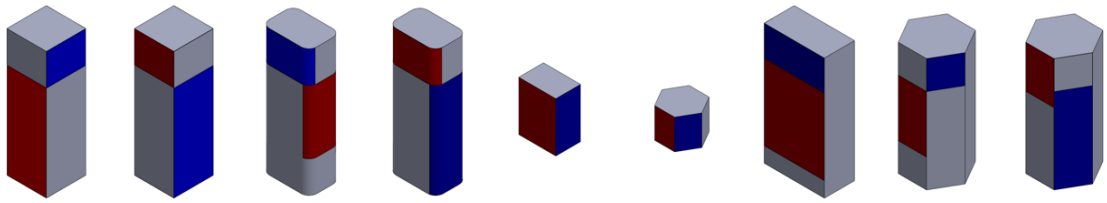

Figure 5.5: Initial and goal region pairs are defined on each object. Red regions indicate the initial regions and blue regions indicate the goal regions.

### 5.2.2 Results

Table 5.1: Experiment Metrics

| Object | Planning time (s) | Execution time (s) | Failure rate |
|---|---|---|---|
| *square prism* | 70.8 | 70.7 | 26% |
| *rectangular prism curved* | 40.1 | 52.3 | 10% |
| *rectangular prism large* | 34.1 | 39.5 | 10% |
| *hexagonal prism tall* | 5.7 | 17.1 | 0% |
| *hexagonal prism short* | 8.5 | 10.0 | 0% |
| *rectangular prism small* | 1.4 | 10.9 | 0% |
| ***All Objects*** | 37.6 | 40.8 | 10% |

To evaluate the performance of our system in region-based manipulation tasks, we compute the ratio of the contact area within the goal to the area of the entire contact region. Computed percentage goal region overlap is reported in Figure 5.6.

Results show that the region-based planning and the in-hand manipulation pipeline can navigate around 70% of the contact region into the goal region for the selected objects. Errors in the final state are mostly due to the open-loop execution of the manipulation primitives, which fails to compensate for the inaccuracies in the modeling and disturbances induced by the unplanned slips during the execu-

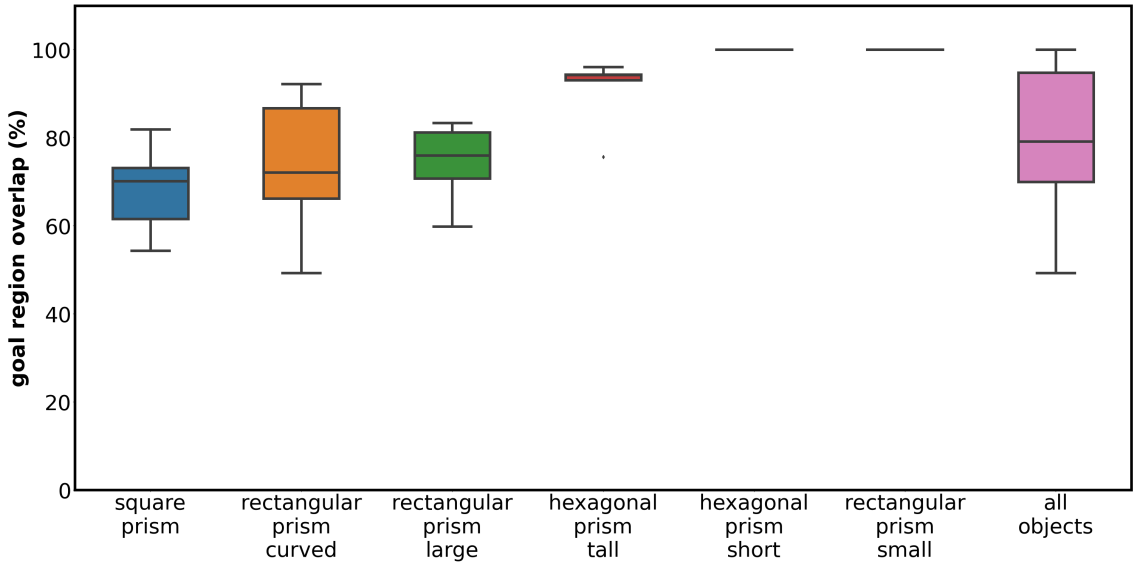


Figure 5.6: Distribution of averaged goal region overlap for left and right fingers on each object. Results include 11 successful trials for square prism, 9 successful trials for rectangular prism curved and rectangular prism large, 5 successful trials for hexagonal prism tall, hexagonal prism short and rectangular prism small for a total of 44 successful trials.

tion. In some cases, unplanned slips lead to dropping the object, which are referred to as failures in Table 5.1.

The performance and the failure rates are highly dependent upon the complexity of the defined regions on the object and corresponding manipulation plans. We define the regions and the manipulation tasks to test the limits of the region-based planning as well as the 3D in-hand manipulation platform. While doing so, we consider the following constraints regarding each object geometry. For prisms with limited height, pivoting and moving up & down are usually invalid actions, as these primitives require some portion of the object along the height to extend out of the grasp. The hexagonal prisms are difficult to slide within the hand, as they are more likely to pivot around the corners. In addition to that, the edge lengths are smaller to keep the object size within an acceptable range, which makes the grasp on hexagonal objects less stable at the fingertips leading to challenges in manip-

ulation at the fingertips. Square and rectangular prisms with proper height are easily manipulated by the available system, thus more complex tasks are preferred on these objects. Square prism, being the object with the most primitive geometry, allows more complex manipulation planning and execution including multiple slides and rotations in sequence with pivoting. However, likelihood of having unplanned slippage during manipulation also increases with the number and complexity of the executed manipulation actions, which caused larger region overlap errors. Curved and large rectangular prisms were subject to less complex sequences including slides with single rotation and pivoting towards the end. Hexagonal prisms and small rectangular prism were not pivoted during the experiments due to the workspace constraints, which reduced the corresponding failure rates and inaccuracies drastically for these objects. The effect of manipulation complexity can also be observed on the planning and execution times provided in Table 5.1.

# Chapter 6

# Discussion and Future Work

While the planning strategy, coupled with the advantages of the variable friction fingers, provides a large operation space, the setup exhibits several mechanical limitations that prevent the execution of manipulation primitives on certain objects or certain states. These limitations sometimes cause inaccuracies or failures during the execution. We accounted for some of these limitations by imposing constraints on the optimization problem (e.g. by introducing some action sequences as workspace constraints, making the system avoid near-failure states). These constraints are parametrically defined inequalities depending on the object and finger geometry. However, the hand was not able to manipulate some object geometries at all, for instance, thin prisms and moderately large objects, which we removed from our experiment set. According to our observations, the major limiting factor was the palm width of the hand. For objects with certain length to width ratios, it is not always possible to exert enough moment to initiate the within-hand rotation motion. Similarly, for some object geometries and palm width, a slight disturbance or modeling mismatch can initiate a rotation, preventing the sliding actions from being executed as planned.

In the future work, we plan to investigate the advantages of having a mechanical design that allows variable palm width. In addition to enabling sliding and rotation for thin prisms, this would also enable the manipulation of larger or smaller objects. However, it is expected that such modifications will increase the system complexity and motion planning requirements. A design addressing the variable palm width mechanism is developed by our external collaborators. As of now, we are working on a manufactured prototype, and testing the capabilities of the mechanism. Two options will be investigated regarding the control of the palm width in the future:

- The algorithm will determine and set the optimal palm width, before initiating the motion planning. Models used within the motion planner will consider the determined palm width.

- Increasing and decreasing the palm width will be defined as manipulation primitives, which can be executed during a manipulation sequence. Corresponding transition models for the planning algorithm needs to be derived.

Another future direction is the adaptation of the planning algorithm to a wider variety of object geometries. This work entails the relaxation of some of the assumptions listed in Section 4.1. An example is the assumption regarding contacts on parallel object surfaces, which restricts the framework to manipulate prismatic objects. Future approach will investigate system states including edge contacts with the objects, in addition to face contact cases.

Another potential improvement on this work is to incorporate closed-loop control and planning to the proposed in-hand manipulation framework. By estimating the object pose within-hand, we plan to account for the modeling errors in variable friction hand kinematics as well as the pivoting kinematics. This might enable us to avoid near failure configurations and reach goal regions with increased accuracy.

Aside from the modifications and additions to mechanical design, modeling and control of this in-hand manipulation framework, there are also some issues to investigate regarding the region-based planning algorithm. As described in Section 4.2.5 and shown in Section 5.2.2, the costs and the heuristic weights are not optimized, which sacrifices the optimality and time complexity for the A* algorithm. These parameters can be tuned further to achieve shorter planning an execution times. Relatedly, a statistical study can be conducted on the relationship between the failure rates, object geometry and selected manipulation primitives to determine a set of parameters that would reduce the number of failures during the execution.

# Chapter 7

# Broader Impacts

Our work offers following technical contributions:

- A formulation and solution for robotic in-hand manipulation that aims to reach specified contact regions on object surfaces
- Increased in-hand manipulation capabilities extending to 3D, enabled via robot fingers with variable friction mechanisms and manipulation strategies leveraging external elements
- Simplistic task execution approach that does not rely on complex sensing and control methods

In addition to the advancements that these contributions might bring in industrial applications and robotics research, they also entail broader impacts involved in domains such as domestic robotics and personalized care, which might be of particular concern to the entire society.

Robotic devices can be used in the personalized care of patients, the elderly, and people with physical limitations and disabilities [80, 81, 82]. Robotic manipulators assist these people in hospitals or household spaces, such as kitchens, bathrooms, or

bedrooms. Some common tasks include helping people to get out of a bed or a chair, use their medicine, visit the bathroom, or fetching personal items and bringing food, etc. As the demand for personalized healthcare increases, it might be preferable to assign even more tasks to robots, and let healthcare personnel monitor and supervise them for more efficient and accessible care. Robots can also be advantageous in treating and assisting people during outbreaks of infectious diseases, where human-to-human contact is to be minimized.

Similarly, robots can be deployed in domestic environments to assist people with daily chores such as cooking, cleaning, laundry, dish-washing, vacuuming, sorting, and organization of items [83, 84]. As robotic technology advances, domestic robots might be more accessible to a wider population. This will allow them to avoid performing such routine tasks after a workday, pursue hobbies, and have more time for leisure. As with healthcare applications, domestic robots can be effective in enabling social distancing during pandemics, when they are deployed for tasks such as shopping and food pick-up.

Most of the tasks mentioned here require some sort of manipulation of objects or usage of handheld tools. Outcomes from our research might facilitate the assignment of these tasks to robotic devices in a couple of ways. Offering a simple but robust solution to the in-hand manipulation problem (e.g., without requiring elaborate sensing and control technologies), our framework might increase the accessibility to robotic systems that are currently in use for domestic purposes and personal care. Through the increased manipulation capabilities discussed and validated in this work, these systems will be able to perform a wide range of tasks in such contexts. The proposed region-based formulation is expected to be useful for the development of alternative planning and control algorithms for in-hand manipulation in the future as well. We believe that it will provide an effective representation for many tasks

including the most common ones in household and healthcare contexts. For example, instead of requiring a cooking robot to grasp a pan and a spatula by making contact with specified points on their handles, our formulation will allow people to define the entire handle as an acceptable grasp region. Similarly, to unscrew a medicine bottle, a caregiver robot will have the flexibility to grasp the bottle body and the bottle cap at any point within the regions valid for the task.

Potential improvements on this work and future directions emerging from our contributions might lead to further advancements in domestic and healthcare robotics. The 3D in-hand manipulation platform utilized in this work can manipulate objects limited to a certain size and geometry, which will be restrictive considering the wide range of objects and tools that are required to be used for the target tasks in domestic and healthcare robotics. Our experiments are performed on prismatic objects, validating that items such as medicine or food boxes can be manipulated through the proposed system. However, there are a lot of utensils and products, such as screwdrivers and disinfectant bottles that have non-prismatic geometries, which require a broader definition of object models. A mechanical design and an accompanying motion planning approach that generalizes well to a wider range of object size, weight, and geometry, will increase the number of tasks that can be performed using this platform. As the targeted environments contain many different objects, it might not be feasible to input accurate geometric models to the platform for each object prior to the deployment of these systems. Furthermore, these objects will vary in brands and models within the households, making it impossible to provide specific and exact models. The size and geometry of a pan or an olive oil bottle might be quite different for different manufacturers and models. If the proposed system is equipped with vision-based sensors and algorithms that can detect and identify objects, resulting robotic devices will be capable of manipulating these

objects without requiring prior object models or human input.

Relatedly, semantic approaches for grasping are becoming available [85, 86, 43, 87]. These approaches allow the robots to recognize and reason about the objects using semantic terms instead of pure mathematical definitions. As a result, robots will become capable of communicating with humans and understanding their commands. When implemented to in-hand manipulation in household environments along with the object detection and identification systems, these approaches might allow robots to perform tasks that can be described by users using everyday language. To emphasize, the region-based formulation will even be more useful in such cases, where the regions on the object surface correspond to semantic descriptions. Robots that can effectively communicate with their users and act according to provided user instructions are expected to increase accessibility, user comfort, and well-being.

The positive impacts of our work and its implications are discussed in detail. However, there are also ethical principles and connected issues associated with this work and its future implementations. As we described the broader impact of this work in human-centered contexts, the resulting technology will be in frequent interaction with humans. These robotic devices will share personal space with humans, communicate with them, and act upon their commands. That is why accountability and transparency should be among the core principles when implementing such technologies. Engineers responsible for developing mechanical designs and algorithms should be aware of the components of the framework that they design. If an item is dropped during in-hand manipulation, the malfunctioning system component, either the mechanism or the algorithm, and corresponding authority should be easily identifiable. Developers should also identify and provide necessary instructions to users, or develop a training procedure if required. Both primary stakeholders that are being assisted by the robotic devices, and the secondary ones that actively use the

devices, such as nurses and doctors, should have access to such specific instructions or training. These steps in turn will allow authorities to assign responsibility for the robot behavior, potential misuse, or accidents. The system should be transparent such that stakeholders are able to find out and understand the reasoning behind the system's decisions and actions. Even the simplistic robotic platform described in this work has to make a variety of different decisions depending on the task and the manipulated object. The algorithms that are used in such platforms can be complex or non-intuitive in nature. For developers, evaluators, and users to operate and assess these devices in a safe and comfortable manner, additional functionalities could be implemented to explain robot behavior in a way suitable for the primary and secondary stakeholders. In addition to these core principles, data privacy issues might arise in both domestic and healthcare contexts, if object detection and identification approaches are implemented in these systems. Detected objects or patient-related data provided to robots might give away information regarding people's medical status or their treatment procedures. Similarly, data on the objects within a household might expose sensitive information about the inhabitants. Users of these systems should be asked for consent regarding the storage and disclosure of their personal data.

To conclude, we believe that our work poses direct and indirect impacts on society through applications in domestic and healthcare robots. On the positive side, this research's outcomes and potential future directions will improve the performance and increase the accessibility of robots in these contexts, contributing to human well-being. However, one should also acknowledge the negative issues that might arise regarding the ethical principles of accountability, transparency, and data privacy, when implementing this technology and building upon the ideas proposed in this work.

# Chapter 8

# Conclusion

In this work, we propose an automatic 3D in-hand manipulation platform that covers a larger workspace than the systems available in the literature and a novel manipulation planning algorithm, which utilizes a region-based formulation that is a more effective representation of the in-hand manipulation tasks. This thesis summarizes the manipulation capabilities enabled by the variable friction fingers and introduces the extrinsic dexterity based strategies used to expand the manipulation workspace. Using the variable friction principle, a quasi-static approach is developed to perform manipulation techniques such as prehensile pushing, gravity exploitation, and pivoting. Relying on the contact mode control achieved through friction modulation at the finger surfaces, manipulation primitives are modeled in purely kinematic-level and executed without elaborate sensing or control.

A region-based in-hand manipulation task is considered and formulated as an optimization problem. Developed region-based motion planning algorithm uses modified A* search to solve the motion planning problem and generate a sequence of manipulation primitives to move the fingers from an initial contact region to a given goal region. The approach is validated in a simulation environment and real-robot

experiments following a standardized in-hand manipulation benchmarking protocol.

A discussion of the system's current capabilities and limitations along with some future research directions is provided. The broader impacts of the work presented in this thesis are investigated in detail.

# Bibliography


[1] J. C. Trinkle and J. J. Hunter, "A framework for planning dexterous manipulation," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pp. 1245–1251 vol.2, 1991.

[2] A. M. Okamura, N. Smaby, and M. R. Cutkosky, "An overview of dexterous manipulation," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, pp. 255–262 vol.1, 2000.

[3] R. R. Ma and A. M. Dollar, "On dexterity and dexterous manipulation," in *2011 15th International Conference on Advanced Robotics (ICAR)*, pp. 1–7, 2011.

[4] N. Chavan-Dafle, M. T. Mason, H. Staab, G. Rossano, and A. Rodriguez, "A two-phase gripper to reorient and grasp," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1249–1255, 2015.

[5] S. Cruciani, C. Smith, D. Kragic, and K. Hang, "Dexterous manipulation graphs," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2040–2047, 2018.

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

[7] B. Sundaralingam and T. Hermans, "Relaxed-rigidity constraints: In-grasp manipulation using purely kinematic trajectory optimization," *Robotics: Science and Systems*, vol. 13, 2017.

[8] O. M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[9] I. Mordatch, Z. Popović, and E. Todorov, "Contact-invariant optimization for hand manipulation," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '12, (Goslar, DEU), p. 137–144, Eurographics Association, 2012.

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

[11] S. Cruciani and C. Smith, "In-hand manipulation using three-stages open loop pivoting," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1244–1251, 2017.

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

[13] H. Liu, P. Meusel, N. Seitz, B. Willberg, G. Hirzinger, M. Jin, Y. Liu, R. Wei, and Z. Xie, "The modular multisensory dlr-hit-hand," *Mechanism and Machine Theory*, vol. 42, no. 5, pp. 612–625, 2007.

[14] S. Funabashi, A. Schmitz, T. Sato, S. Somlor, and S. Sugano, "Robust in-hand manipulation of variously sized and shaped objects," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 257–263, 2015.

[15] A. Holladay, R. Paolini, and M. T. Mason, "A general framework for open-loop pivoting," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3675–3681, 2015.

[16] N. Rojas, R. R. Ma, and A. M. Dollar, "The gr2 gripper: An underactuated hand for open-loop in-hand planar manipulation," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 763–770, 2016.

[17] R. R. Ma, A. Spiers, and A. M. Dollar, "M2 gripper: Extending the dexterity of a simple, underactuated gripper," in *Advances in Reconfigurable Mechanisms and Robots II* (X. Ding, X. Kong, and J. S. Dai, eds.), (Cham), pp. 795–805, Springer International Publishing, 2016.

[18] A. J. Spiers, B. Calli, and A. M. Dollar, "Variable-friction finger surfaces to enable within-hand manipulation via gripping and sliding," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4116–4123, 2018.

[19] J. A. Raj, "Autonomous vision-based control for within-hand manipulation using variable friction fingers," Master's thesis, Robotics Engineering Department, Worcester Polytechnic Institute, 2020.

[20] Q. Lu, A. B. Clark, M. Shen, and N. Rojas, "An origami-inspired variable friction surface for increasing the dexterity of robotic grippers," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2538–2545, 2020.

[21] J. K. S. Matthew T. Mason, *Robot Hands and the Mechanics of Manipulation*. Cambridge, MA: MIT Press, 1985.

[22] S. Jacobsen, E. Iversen, D. Knutti, R. Johnson, and K. Biggers, "Design of the utah/m.i.t. dextrous hand," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 1520–1532, 1986.

[23] H. Mnyusiwalla, P. Vulliez, J. Gazeau, and S. Zeghloul, "A new dexterous hand based on bio-inspired finger design for inside-hand manipulation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 6, pp. 809–817, 2016.

[24] C. D. Santina, C. Piazza, G. Grioli, M. G. Catalano, and A. Bicchi, "Toward dexterous manipulation with augmented adaptive synergies: The pisa/iit softhand 2," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1141–1156, 2018.

[25] G. P. Kontoudis, M. Liarokapis, and K. G. Vamvoudakis, "An adaptive, humanlike robot hand with selective interdigitation: Towards robust grasping and dexterous, in-hand manipulation," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pp. 251–258, 2019.

[26] C. Konnaris, C. Gavriel, A. A. C. Thomik, and A. A. Faisal, "Ethohand: A dexterous robotic hand with ball-joint thumb enables complex in-hand object manipulation," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pp. 1154–1159, 2016.

[27] D. Keymeulen and C. Assad, "Investigation of the harada robot hand for space," in *IEEE RAS Internation Conference on Humanoid Robots University International Conference Center, Tokyo, Japan*, 2001.

[28] L. B. Bridgwater, C. A. Ihrke, M. A. Diftler, M. E. Abdallah, N. A. Radford, J. M. Rogers, S. Yayathi, R. S. Askew, and D. M. Linn, "The robonaut 2 hand - designed to do work with tools," in *2012 IEEE International Conference on Robotics and Automation*, pp. 3425–3430, 2012.

[29] J. Ueda, M. Kondo, and T. Ogasawara, "The multifingered naist hand system for robot in-hand manipulation," *Mechanism and Machine Theory*, vol. 45, no. 2, pp. 224–238, 2010.

[30] A. Kochan, "Shadow delivers first hand," *Industrial Robot*, vol. 32, no. 1, pp. 15–16, 2005.

[31] J. Zhou, J. Yi, X. Chen, Z. Liu, and Z. Wang, "Bcl-13: A 13-dof soft robotic hand for dexterous grasping and in-hand manipulation," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3379–3386, 2018.

[32] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 161–185, 2016.

[33] S. Abondance, C. B. Teeple, and R. J. Wood, "A dexterous soft robotic hand for delicate in-hand manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5502–5509, 2020.

[34] L. U. Odhner, L. P. Jentoft, M. R. Claffee, N. Corson, Y. Tenzer, R. R. Ma, M. Buehler, R. Kohout, R. D. Howe, and A. M. Dollar, "A compliant, underactuated hand for robust manipulation," *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 736–752, 2014.

[35] K. Batsuren and D. Yun, "Soft robotic gripper with chambered fingers for performing in-hand manipulation," *Applied Sciences*, vol. 9, no. 15, 2019.

[36] N. Rahman, L. Carbonari, M. D'Imperio, C. Canali, D. G. Caldwell, and F. Cannella, "A dexterous gripper for in-hand manipulation," in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 377–382, 2016.

[37] J. Spiliotopoulos, G. Michalos, and S. Makris, "A reconfigurable gripper for dexterous manipulation in flexible assembly," *Inventions*, vol. 3, no. 1, 2018.

[38] C. M. McCann and A. M. Dollar, "Design of a stewart platform-inspired dexterous hand for 6-dof within-hand manipulation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1158–1163, 2017.

[39] L. U. Odhner and A. M. Dollar, "Dexterous manipulation with underactuated elastic hands," in *2011 IEEE International Conference on Robotics and Automation*, pp. 5254–5260, 2011.

[40] R. R. Ma and A. M. Dollar, "An underactuated hand for efficient finger-gaiting-based dexterous manipulation," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pp. 2214–2219, 2014.

[41] *In-Hand Manipulation Primitives for a Minimal, Underactuated Gripper With Active Surfaces*, vol. Volume 5A: 40th Mechanisms and Robotics Conference of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 08 2016. V05AT07A072.

[42] W. G. Bircher, A. M. Dollar, and N. Rojas, "A two-fingered robot gripper with large object reorientation range," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3453–3460, 2017.

[43] H. Liu, L. Zhao, B. Siciliano, and F. Ficuciello, "Modeling, optimization, and experimentation of the paragripper for in-hand manipulation without parasitic rotation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3011–3018, 2020.

[44] B. Ward-Cherrier, N. Rojas, and N. F. Lepora, "Model-free precise in-hand manipulation with a 3d-printed tactile gripper," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2056–2063, 2017.

[45] J. Amend and H. Lipson, "The JamHand: Dexterous Manipulation with Minimal Actuation," *Soft Robotics*, vol. 4, no. 1, pp. 70–80, 2017.

[46] V. Tincani, M. G. Catalano, E. Farnioli, M. Garabini, G. Grioli, G. Fantoni, and A. Bicchi, "Velvet fingers: A dexterous gripper with active surfaces," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1257–1263, 2012.

[47] N. Govindan and A. Thondiyath, "Design and Analysis of a Multimodal Grasper Having Shape Conformity and Within-Hand Manipulation With Adjustable Contact Forces," *Journal of Mechanisms and Robotics*, vol. 11, 07 2019. 051012.

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

[49] N. Chavan-Dafle, R. Holladay, and A. Rodriguez, "Planar in-hand manipulation via motion cones," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 163–182, 2020.

[50] F. E. Viña B., Y. Karayiannidis, K. Pauwels, C. Smith, and D. Kragic, "In-hand manipulation using gravity and controlled slip," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5636–5641, 2015.

[51] S. Stepputtis, Y. Yang, and H. Ben Amor, "Extrinsic dexterity through active slip control using deep predictive models," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3180–3185, 2018.

[52] M. Costanzo, G. De Maria, and C. Natale, "Slipping control algorithms for object manipulation with sensorized parallel grippers," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7455–7461, 2018.

[53] M. Cherif and K. K. Gupta, "3d in-hand manipulation planning," in *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*, vol. 1, pp. 146–151 vol.1, 1998.

[54] J. Saut, A. Sahbani, S. El-Khoury, and V. Perdereau, "Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2907–2912, 2007.

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

[56] S. Cruciani, K. Hang, C. Smith, and D. Kragic, "Dual-arm in-hand manipulation and regrasping using dexterous manipulation graphs," 2019.

[57] S. Cruciani, H. Yin, and D. Kragic, "In-hand manipulation of objects with unknown shapes," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8848–8854, 2020.

[58] W. G. Bircher, A. S. Morgan, K. Hang, and A. M. Dollar, "Energy gradient-based graphs for planning within-hand caging manipulation," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2462–2467, 2019.

[59] B. Calli, A. Kimmel, K. Hang, K. Bekris, and A. Dollar, "Path planning for within-hand manipulation over learned representations of safe states," in *Proceedings of the 2018 International Symposium on Experimental Robotics* (J. Xiao, T. Kröger, and O. Khatib, eds.), (Cham), pp. 437–447, Springer International Publishing, 2020.

[60] B. Calli, K. Srinivasan, A. Morgan, and A. M. Dollar, "Learning modes of within-hand manipulation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3145–3151, 2018.

[61] L. Han, Y. S. Guan, Z. X. Li, Q. Shi, and J. C. Trinkle, "Dextrous manipulation with rolling contacts," in *Proceedings of International Conference on Robotics and Automation*, vol. 2, pp. 992–997 vol.2, 1997.

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

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

[64] N. Daoud, J. Gazeau, S. Zeghloul, and M. Arsicault, "A real-time strategy for dexterous manipulation: Fingertips motion planning, force sensing and grasp stability," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 377–386, 2012. Autonomous Grasping.

[65] U. Scarcia, K. Hertkorn, C. Melchiorri, G. Palli, and T. Wimböck, "Local online planning of coordinated manipulation motion," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6081–6087, 2015.

[66] N. Chavan-Dafle and A. Rodriguez, "Sampling-based planning of in-hand manipulation with external pushes," in *Robotics Research* (N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti, eds.), (Cham), pp. 523–539, Springer International Publishing, 2020.

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

[68] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," 2018.

[69] A. Nagabandi, K. Konolige, S. Levine, and V. Kumar, "Deep dynamics models for learning dexterous manipulation," in *Proceedings of the Conference on Robot Learning* (L. P. Kaelbling, D. Kragic, and K. Sugiura, eds.), vol. 100 of *Proceedings of Machine Learning Research*, pp. 1101–1112, PMLR, 30 Oct–01 Nov 2020.

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

[71] T. Li, K. Srinivasan, M. Q. H. Meng, W. Yuan, and J. Bohg, "Learning hierarchical control for robust in-hand manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8855–8862, 2020.

[72] A. Sintov, A. Kimmel, K. E. Bekris, and A. Boularias, "Motion planning with competency-aware transition models for underactuated adaptive hands," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7761–7767, 2020.

[73] R. Ma and A. Dollar, "Yale openhand project: Optimizing open-source hand designs for ease of fabrication and adoption," *IEEE Robotics Automation Magazine*, vol. 24, no. 1, pp. 32–40, 2017.

[74] P. N. Stuart Russell, *Artificial Intelligence: A Modern Approach*. Prentice Hall, Pearson Education, Inc., 1994.

[75] K. Crane, C. Weischedel, and M. Wardetzky, "The heat method for distance computation," *Communications of the ACM*, vol. 60, pp. 90–99, Oct. 2017.

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

[77] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Sendai, Japan), pp. 2149–2154, Sep 2004.

[78] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, (Kobe, Japan), May 2009.

[79] S. Cruciani, B. Sundaralingam, K. Hang, V. Kumar, T. Hermans, and D. Kragic, "Benchmarking in-hand manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 588–595, 2020.

[80] RIKEN, "The strong robot with the gentle touch," 2015. https://www.riken.jp/en/news_pubs/research_news/pr/2015/20150223_2/, Retrieved on 2021-04-30.

[81] AVATARMIND, "iPal Platform," 2017. https://www.ipalrobot.com/ipal-tm-platform, Retrieved on 2021-04-30.

[82] A. M. Okamura, M. J. Matarić, and H. I. Christensen, "Medical and health-care robotics," *IEEE Robotics Automation Magazine*, vol. 17, no. 3, pp. 26–37, 2010.

[83] U. Reiser, C. Connette, J. Fischer, J. Kubacki, A. Bubeck, F. Weisshardt, T. Jacobs, C. Parlitz, M. Hägele, and A. Verl, "Care-o-bot® 3 - creating a product vision for service robot applications by integrating design and technology," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1992–1998, 2009.

[84] M. Cakmak and L. Takayama, "Towards a comprehensive chore list for domestic robots," in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 93–94, 2013.

[85] H. Dang and P. K. Allen, "Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1311–1317, 2012.

[86] E. Jang, S. Vijayanarasimhan, P. Pastor, J. Ibarz, and S. Levine, "End-to-end learning of semantic grasping," *CoRR*, vol. abs/1707.01932, 2017.

[87] R. Detry, J. Papon, and L. Matthies, "Task-oriented grasping with semantic and geometric scene understanding," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3266–3273, 2017.