Direct-drive Hands: Making Robot Hands Transparent and Reactive to Contacts

Ankit Bhatia

CMU-RI-TR-22-16

May 11, 2022

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Matthew T. Mason, *Co-chair*
Aaron M. Johnson, *Co-chair*
Zachary Manchester
Scott A. Bortoff, *Mitsubishi Electric Research Labs*

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics.

To my immensely talented sisters,
Arushi, Janavi, Josita, Manasvini, Mrinalini, Pratishtha, Yastika.
You are the future.

Abstract

Industrial manipulators and end-effectors are a vital driver of the automation revolution. These robot hands, designed to reject disturbances with stiffness and strength, are inferior to their human counterparts. Human hands are dexterous and nimble effectors capable of a variety of interactions with the environment.

Through this thesis we wish to answer a question: How can we make robot grippers better at interacting with the environment? A key idea essential for supporting a wide range of interactions with the environment is transparency: the efficient transfer of force and motion between the robot and the task. In search for an answer, we explore the key developments that allow robots to be reactive to contacts. We analyze the space of actuators to determine parameters that affect transparency and define a metric to characterize transparency by the collision reflex response: the impulse transferred to a rigid, fixed object in a collision.

We show that direct-drive is a favorable transmission choice for improving transparency. Recent advances in motor technology have triggered a resurgence in direct-drive and quasi-direct-drive systems. The MIT Cheetah, Ghost Robotics' Minitaur, and Agility Robotics' Cassie are all examples of successful legged robots that have improved transparency without compromising their torque specifications.

We have developed the DDhand, a transparent direct-drive gripper inspired by the direct-drive robots from the 80's. The DDHand is a 4 degree of freedom, two fingered gripper with a parallel five bar linkage to keep link reflected inertias low. The gripper improves on speed, bandwidth and mechanical simplicity over other comparable grippers.

Transparency in robotic grippers allows for performance gains in contact-rich and dynamic behaviors and new approaches in intrinsic contact sensing. We present the smack-and-snatch grasp: a contact-informed grasping behavior that can grasp an object on a table of unknown height in a rapid arm motion. We show a velocity-based contact localization algorithm that can localize contact locations using only position and velocity sensing. Finally, we demonstrate the viability of the DDHand in industrial applications.

Acknowledgments

Firstly, I would like to acknowledge my advisors, Matthew T. Mason and Aaron M. Johnson who have given me invaluable advice on all things robotics and otherwise. Thanks for allowing me the freedom to pitch my own ideas and helping me frame them into a concrete research direction. I would like to thank my committee, Zachary Manchester and Scott Bortoff for asking the hard questions. Scott, thank you for the countless discussions on robot collisions at MERL. Ralph Hollis, thank you for being an amazing teacher at the Microdynamics Systems Lab. You deserve the credit for most of what I know about robots. David Bourne and Howie Choset, thank you for your support over the years.

To all the members and alumni of the RoboMechanics Lab: Joe Norby, Catherine Pavlov, Sean Wang, Joe Payne, Nathan Estansi, Justin Yim; and the Manipulation Lab: Jiaji Zhou, Robbie Paolini, Zhenzhong Jia, Yifan Hou, Xianyi Cheng, Eric Huang, Jon King, Pragna Mannam, Alex Volkov, Reuben Aronson, Gilwoo Lee, Arnav Gupta, Yuemin Mao and Amy Santoso, you have all been amazing collaborators, colleagues and teachers. Thank you Jean Harpley, for keeping us and the lab out of trouble.

To Rogerio Bonatti, Alex Spitzer, Roberto Shu, Humphrey Hu, Michael Shomin, Adithya Murali, Ashwin Khadke, Dominik and Cornelia Bauer, Eugene Fang, Dinesh Reddy, and all the other friends I've made along the way, thank you for being amazing co-passengers on this journey.

To the Rocket League champions: Senthil Purushwalkum and Achal Dave; the Burger Bandits: Brian Okorn, Leonid Kesselman, Adam Harley and Benjamin Newmann; and my tennis partners: Vishal Dugar, Anirudh Vemula, Eric Huang, Manan Shah and Allie Del Giorno, thank you for bringing in some balance to my days. I will always remember our late night video game sessions, arbitrary burger ratings and time spent on tennis courts around Pittsburgh.

To my manipal cohort who made the trip with me to Pittsburgh: Riddhit Mitra, Shruti Karat, Nishita Sinha, and Rishabh Brajabasi, thanks for being there to reminisce about easier times. Thank you Surya Dev Aggarwal, Mahesh Pai, and Amrita Sahu for being the absolute best roommates one could ask for. Surya, know that without your sustained motivation, this thesis would have taking much longer to write. Thank you Rohit Garg, for being on my team for the last twenty years. It was great to have such a familiar face across the road. I will miss our midnight fast-food runs.

Finally, I would like to acknowledge my family's support, patience and love. To my mother, Naina, father, Ravi, and sisters Arushi and Manasvini thank you for being the amazing cheering squad you have been throughout the process. Thank you all for sharing the happiness of my successes, the sorrows of my failure and the hardship of my challenges.

Funding

This work was supported in part by a National Science Foundation Grant (IIS-1813920), Carnegie Mellon University's Presidential Fellowship, and other industrial sponsors.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

What are hands for? If your use is limited to grasping, then you will pick a gripper designed like a vise: strong, resistant to large disturbances and can support a wide range of objects. But hands are used for much more than that. Human hands are capable of a wide range of behaviors like pushing, pulling, throwing, catching, etc. A better answer then is that hands are for interacting with the environment. This thesis explores improving the performance of interactions of the end-effector and the environment by way of improving their sensing and reactionary capabilities.

The vise-like industrial robot gripper has been pivotal in automating today's manufacturing industry. These grippers are essential workers in the manufacturing of almost every modern piece of technology from cell phones and computers to cars and aircraft. Despite their ubiquity, their utility is fairly limited. With targeted applications like pick-and-place and peg-in-hole assembly, these are designed to operate largely in collision free environments with contact events being carefully controlled. The focus is on adding stiffness and disturbance rejection using large gear ratios. In contrast, humans have evolved to be versatile manipulators capable of a wide range of interactions. Our dexterous and nimble hands welcome collisions with the environment, often using contact events for efficiency gains. This dexterity comes at the cost of a reduced payload capacity compared to their industrial counterparts.

A key design feature for supporting a wide range of interactions is transparency: the efficient transmission of force and motion between the robot and the environment. Ideas of transparency stem from the analysis of teleoperation systems. Teleoperation systems are treated as transmissions and the focus is on input-output impedance matching. We adapt this idea of transparency for a robot gripper: for an ideal robot, one would want the robot joints to exactly mimic the command signal (*actuator transparency*) and any external applied forces to reflect in the feedback instantaneously (*perception transparency*). Elements like communication latency, robot dynamics, gearbox scaling, compliance from springs and cable drives etc. negatively affect this transparency.

The design of robot hands has not specially considered transparency as a design parameter. Even so, recent years have seen an improvement in the perception transparency of robot hands using active and passive compliance schemes. The design of compliance in robot hands and other actuators follows a circuitous route. This approach generally starts with a compliant source (electric motor), adds stiffness with a gearbox, and then adds passive or active springs resulting in a compliant source. Besides improving the perception transparency of the system, this approach adds mechanical complexity and reduces bandwidth and controllability.

The thesis takes inspiration from pioneering work on direct-drive (DD) robots by Asada et al. [5]. Direct-drive transmissions improve bidirectional transparency (actuation and perception). This approach forgoes smaller motor and high-ratio gearbox combinations for adequately high torque motors directly coupled to the joint. This removes the mechanical elements responsible for loss of transparency and allows for higher bandwidth control without introducing underactuation. Until recently, direct-drive actuation has been primarily limited to specialized applications due to reduced payloads and unfavorable scaling. Improvements in torque density of Brushless Direct Current (BLDC) motors and high efficiency drive electronics are making the case for mainstream direct-drive transmissions. Legged robots like the Minitaur [63] and the Cheetah [115] are employing direct-drive (unit gear ratios) and quasi-direct-drive (QDD) (low gear ratio) transmissions to improve transparency.

This thesis focuses on the adaptation of transparent actuation for contact-rich manipulation. We show performant behaviors that are made possible with direct-drive grippers. The goals of the thesis are reflected in the following thesis statement:

> *Transparent actuation is key for robotic hands to support a wider range of interactions than just grasping. Improving the transparency of robotic hands through direct-drive actuation improves their ability to sense and react to contact.*

1.1 Contributions of the thesis

1. An analysis of transmissions in robotic grippers and the effect of reflected inertia and compliance on transparency.
2. Characterization of the transparency of a robotic system by measuring the collision reflex – the total impulse transaction during an unexpected collision which the system is actively trying to mitigate.
3. A prototype design and control architecture for a direct-drive gripper that improves transparency.
4. Demonstration of the benefits of transparent grippers through contact localization using velocity constraints

5. Demonstration of dynamic manipulation behaviors like move-until-touch and smack-and-snatch grasping.
6. Demonstration of the utility of the prototype direct-drive gripper in industrial automation.

1.2 Organization

Chapter 2: Background: We discuss the state of the art in compliance, direct-drive actuation, hand design and control and transparency.

Chapter 3: Transparency and Collision Reflexes: In this chapter we analyze transparency through simple models of actuators and robots. To begin we investigate the benefits of direct-drive transmissions and how bidirectional transparency can be achieved. Then we show how direct-drive actuation improves the speed, transparency and force bandwidth while keeping the design mechanically simple. We discuss common scaling laws and their affect on transparency. Finally, we define a metric that captures the collision reflex capability of a robot using total impulses transferred during a collision event. Using this metrix, we analyze the effect of actuator scaling and robot configuration.

Chapter 4: Design of a direct-drive hand: We describe the design of a direct-drive hand prototype with two degrees of freedom (DOF) of direct-drive actuation per finger coupled with a five bar linkage transmission. We explore the control architecture for this system and its implementation using custom motor drive electronics and Robot Operating System (ROS). We also introduce a Dexterous DDHand with simpler parallel jaw kinematics.

Chapter 5: Move-until-touch Behaviors We implement the smack-and-snatch behavior that exploits all the benefits of transparent actuation in manipulation. We then look at the collision reflex on physical systems comparing single Degrees of freedom (DOF) actuators and end-effectors.

Chapter 6: Contact localization using Velocity Constraints In this chapter, we detail an intrinsic tactile sensing algorithm that uses joint position and velocity to identify contacts on a link. The algorithm needs the linkage to be backdrivable which is a direct result of perception transparency.

Chapter 7: Applications of the DDHand in Small Parts Assembly We explore the viability of using the DDHand in industrial applications. Some favorable initial results are shown in the application of the Dexterous DDHand in Small Parts Assembly.

Chapter 8: Future Work and Conclusion We suggest some future directions for the work described in this thesis and conclude with a few closing remarks.

1.3 Publication Note

Analysis of actuation (Section 3.1), design of the prototype gripper (Chapter 4) and the smack-and-snatch grasp (Section 5.1) were introduced in [11]. A patent application [12] has been filed for the design of the DDHand (Chapter 4). Contact localization using Velocity Constraints (Chapter 6) is joint work with Sean Wang and appeared in [129]. Work on the collision reflex metric (Chapter 3) is in preparation for publication at IEEE Transactions on Robotics. Applications of DDHand in small parts assembly (Chapter 7) is joint work with Arnav Gupta, Yuemin Mao, Xianyi Cheng, Jonathan King and Yifan Hou. This work is being revised for IEEE Robotics and Automation Letters.

Chapter 2

Background

2.1 Compliance

Most robotic manipulators are intended to produce programmed motions accurately, regardless of the forces encountered. In short, they are stiff and obtain high joint torques by using high gear ratios. High gear ratios also lead to high reflected inertias. That means that the inertia perceived by an external observer would include the motor rotor inertia scaled by the gear ratio squared, which dominates the inertia of the arm structure by a considerable margin. That is all fine if the goal is to produce a programmed motion, regardless of forces encountered. It is a great approach for industrial robotics, where there is virtually no role for online intelligence. All the intelligence is offline, which in turn implies that the task environment must conform to offline expectations. Objects must be of predictable shape and in predictable locations, often referred to as "structured environments."

The limitations of the stiff programmed motion approach have long been recognized, and two approaches have been pursued to produce compliant motion: (1) active compliance using a force/torque sensor, as proposed in, e.g., [16, 42, 82, 101, 132]; and (2) passive compliance between the actuators and the load. The passive compliant mechanism could take many forms, such as a spring [99, 131], a differential [15, 27], or a breakaway clutch [124]. One example is the Remote-Center Compliance (RCC) [131]. Another approach, Series-Elastic Actuators (SEA) [99], involves placing one or more springs directly in series with each actuator.

There is much to say about the merits and limitations of both approaches [138], but the inescapable observation is that both approaches start with a compliant source, the electric motor, then add structure and control to make it stiff, and then add more structure and more control to make it compliant again. The alternative is both obvious and appealing: discard the complexity, weight and expense of all the additional stuff, and rely on the native compliance of the actuator: direct-drive.

2.2 Direct-drive Actuation

The appeal of direct-drive has been recognized for at least 30 years [5, 30, 109] but it was not practical at the time. Factory automation dominated the commercial applications, and motors lacked sufficient torque. As applications broaden to include unstructured environments and human interaction, and as actuator torque improves, direct-drive is an inevitable addition to the available manipulation systems.

The earliest investigation of direct-drive actuation was the work of Takeo Kanade and Harry Asada [5]. Applications of direct-drive actuation in end-effector design were subsequently explored [18, 30, 62, 76, 109] but failed to pick up momentum due to lacking torque density in motors. There were products with direct-drive actuation in the 1980s but perhaps the first harbingers of substantial commercial application are now appearing, first in specialized applications [2], and subsequently in general purpose manipulators [32, 33]. As motor technology has improved, the well-known advantages of direct-drive (and low gear ratios more generally) has led to a greater interest in the locomotion community [31, 51, 61, 63, 114]. In particular, [63] contains an overview of the advantages and disadvantages of direct-drive, applied to locomotion.

A discussion of the scaling of actuators and transmissions is essential to understand the applicability of direct-drive actuation for a particular application. There have been many suggested scaling laws for gearboxes and actuators [63, 114, 115, 118]. Two popular scaling laws proposed by Haddadin et al. [37] and Sangbae Kim et al. [114, 130] are discussed and empirically compared in [107]. A discussion of transmission scaling laws can be found in [106].

2.3 Hand design and control

There are two main lines of hand technology development: commercial and research. Commercial applications are dominated by pneumatic grippers, simple binary devices lacking the controllability and instrumentation required for reactive manipulation. Several commercial electrically actuated grippers are available, using high gear ratios to develop large clamping forces [89]. The other line of development is robotics research, which has focused on more complex hands. Although there is a long history of research in robotic hands [13], direct-drive has not played a significant role. Supplementing gearboxes with series-elastic actuation and strain gauges have been popular actuation modes for robot effector design. Examples include the Ishikawa Hand [91], the Yale Hand [80] and the recent Dynamic Observable Contact Hand [58]. Detailed discussion of a few related designs follows and a comprehensive survey of robot hand designs can be found in [3, 97], and a discussion of actuation modes and transmissions can be found in [123].

Relevant Hand Designs

We highlight gripper design ideas relevant to the idea of improving transparency and collision reflexes. We identify the general approach in the design of high-speed and compliant hands and cite a few examples.

High-speed Hands The 8 DOF, four-fingered Ishikawa hand [91] is a flagship of this category. Each joint is designed with a small motor coupled with a custom designed harmonic drive and a strain gauge-based joint torque sensor. The hand can produce about 4 N of gripping force at the finger tip. This hand is coupled with a 1000 Hz vision system to enable highly dynamic skills like dribbling, throwing and catching. It should be noted that the reactive capabilities of this gripper are not well understood. Extensions of this design have been developed with the Dynamic Observable Contact Hand [58] and the high-speed and high-precision robotic hand for micromanipulation [112]. High-speed hands geared towards industrial applications mostly target the micromanipulation industry. The IBM hummingbird device [60] is an example of an early high-speed direct-drive end-effector. A counterweight balanced five-bar linkage driven by voice coil actuators was used with a z-axis probe for circuit testing.

Compliant Hands A survey of robot grippers since 1992 conducted by Piazza et al. [97] throws light on a steady rise in compliant actuation in robot hands. This actuation can be introduced through either active, passive compliance or soft materials. Adding compliance to a joint adds underactuation to the system but improves sensing and reduces impact forces in some cases. Examples of active compliance in robotic hands include the Ishikawa hand [91] and the Schunk gripper [113]. These grippers use strain gauges at the joint or fingertip to close the loop on force. Although these strain gauges improve the force sensing bandwidth, the closed-loop bandwidth is generally low due to high mechanical inertia. The series-elastic approach trades the stiffer compliance elements in strain gauges with softer springs. Encoders measuring the deflection in the spring with a known stiffness provide the force feedback. The softer springs lower the sensing bandwidth while also reducing the reflected inertias during collisions. Lower reflected inertias make this approach suitable for applications where human-robot interactions are needed. Grippers employing series-elasticity add degrees of underactuation that degrades the positional accuracy of the gripper. This problem can be addressed by instrumenting the compliance. Recent advances in variable stiffness actuators allow for the tunable spring constants at the end-effector [66, 74, 87] that can circumvent some of the issues with series-elastic compliance. Variable stiffness is achieved using antagonistic springs, magnets or flextures. The variable stiffness allows for a continuum of behaviors mimicking either rigid or compliant actuators.

2.4 Transparency

We borrow and apply the term "transparency" throughout this thesis. The term originated in teleoperation, where it means that the operator feels as if directly present in the task. Teleoperation researchers [40, 44, 73, 110] adopted models and analysis that transform the intuitive "feeling present" notion into conditions on the transmission of force and velocity between operator and task. The direct-drive hand adopts the same idea: that the conditions that provide the "feeling present" experience in teleoperation should also apply to couple the robotic end-effector to the controller. The key is efficient bidirectional transmission of information carried by force and velocity signals.

A transparent teleoperation system transmits forces from the input to the output instantaneously and vice versa. In teleoperation it is generally reduced to matching the impedance felt by the operator (Z_h) to that of the environment (Z_e) [73]:

$$Z_e = Z_h. \tag{2.1}$$

Z_e and Z_h are related to force and velocity in the environment (F_e, V_e) and at the operator(F_h, V_h) as

$$F_h = Z_h V_h,\ F_e = Z_e V_e. \tag{2.2}$$

A stronger condition is a perfectly transparent transmission where $F_e = F_h$ and $V_e = V_h$.

Let's consider a simple transmission: a spur drive gearbox. Here, $F_e = NF_h$ and $V_h = NV_e$. Although the input and output impedance are equal, gearbox does not meet the stronger condition for perfect transparency. On the contrary, a direct-drive transmission is perfectly transparent.

Over the years, a few metrics have been introduced that describe the transparency properties of the system. Most metrics only analyze the effect of the system's structure on the transparency of the system. These include the Inertia, Ellipsoid of Gyration [43], Dynamic Manipulability and Impact Ellipsoids [128] and the effective mass belted ellipsoid [65] and the more recent impact mitigation factor [130]. The feel-cage task presents a formal definition of transparency which covers the effect of structure, control and sensing of a system. These metrics are discussed in further detail in Chapter 3.

Chapter 3

Transparency and the Collision Reflex Metric

Introduction

Robots are moving out of the structured environment of a factory and into more complicated environments like warehouses and homes. In these spaces, we would like robots to do tasks like small parts assembly, cooperative manipulation, and putting away the dishes. But these new applications in cluttered spaces require more delicate intentional interaction with objects as well as safe unexpected contact with the world. Goertz [34] explains the problem well: "General-purpose manipulation consists essentially of a series of collisions with unwanted forces, the application of wanted forces and the application of desired motions. The collision forces should be low and any other unwanted forces should be small." In this chapter we aim to better characterize design and control of robots for a lighter touch.

There are many ways to mitigate the effect of a collision. In manufacturing applications, robots tend to approach a possible contact event at slower speeds than when operating in free space to minimize impulses, e.g. [54, 72]. When more precision is needed, touch-based localization can be used to approach with a guarded move, i.e. a "move until touch" behavior [82, 133].

Higher velocity guarded moves can cause large impacts and damage the work piece or the robot itself. Another challenge emerges when considering collaborative robots and their ability to handle contacts with humans. These robots cannot always rely on exteroceptive sensors to anticipate the presence of a human in their motion path. Similarly, in robotic locomotion, when a walking robot impacts the leg on the ground, high impacts will cause mechanical noise while walking and lower efficiency.

Being able to manage these impacts through design and control advances can lead to better performing behaviors. Haddadin [38] looks at design and control strategies with the KUKA LWR to make industrial manipulators safe in human-robot collisions through

reactive strategies. In Chapter 5 (also [11]), we demonstrate high-speed grasping behaviors with low-inertia, high bandwidth grippers. Low inertia fingers are also useful for tactile exploration, contact localization and shape recovery [75, 129].

The transparency [64, 114] of a system plays a role in how it manages the impulse during an unexpected collision event. Transparency captures the efficiency of transmission of force and motion between the task and the control agent. This idea of transparency is derived from teleoperation research where it is formally defined as the ratio of operator and task impedance [73]. No such formal definition exists for closed-loop robotic systems. Instead, we discuss the transparency properties of a system by looking at metrics that describe the inertial properties [114] and closed-loop responses [64] of the system. Although these metrics have been useful for optimizing the design and control of robots to minimize impacts, no single metric captures the combined effect of inertia, sensing, and control.

We propose the collision reflex metric as a way to describe the transparency properties of the system. Haddadin [38] first discussed the collision reflex as a way to understand the response of a robot during an unexpected collision event. We extend this idea and analyze the total impulse transferred in such a collision event and discuss the impulsive contribution of the structure, sensing, and control. We analyze both a simple, one-dimensional example and extend this to higher dimensional systems, e.g. a two-link manipulator in the plane.

In Chapter 5 we describe experiments to compare the total impulse transferred in collision of physical systems and assess the collision reflex metric for various actuator configurations and COTS grippers.

In summary, the contributions of the chapter are: 1) A metric which quantifies the collision reflex capability of a robotic system by measuring the total impulse imparted to a fixed rigid object; 2) An extension of this collision metric to cover cases that can fit the general manipulator equation and higher dimensional systems; and 3) Analysis of the effect of control, structure, and sensing on the total impulse transaction in a collision.

3.1 Models of Actuator Transmission

Actuation schemes for servoed robotic grippers can be broadly divided into three categories:

- geared actuation with series-elastic elements;
- geared actuation with strain-gauges;
- direct-drive actuation.

The demarcation between these categories is blurry; they lie on a continuum as shown in Fig. 3.3. For example, a strain gauge is essentially a stiffer series elastic element and an ideal direct-drive can be approximated by a series-elastic actuation scheme with infinite spring stiffness and a unit gear ratio. To elucidate the inherent differences between these schemes, we use a general model of an actuator and transmission that includes each as a special case (Fig. 3.1). From this model, we examine the equations of motion, scaling of the

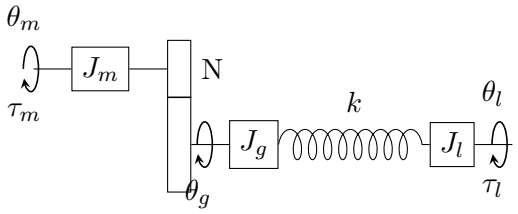


Figure 3.1: General model of an actuator. This model is used to describe a continuum of actuator configurations varying from direct-drive to series-elastic with gearing.

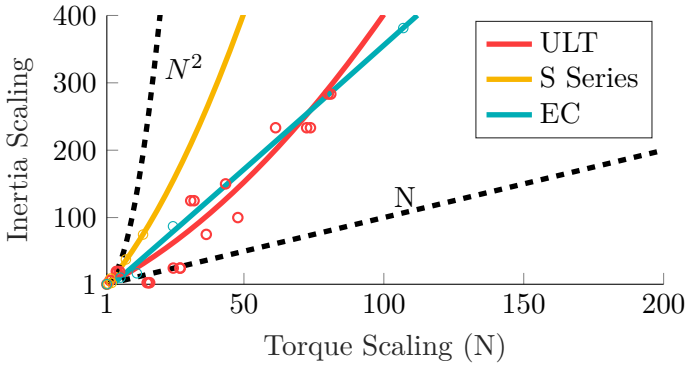


Figure 3.2: Scaling laws for three commercially available motor series are compared against the N^2 scaling law for gearboxes. The ULT series motors are manufactured by Celera Motion, the S series by Aerotech and the EC motors by Maxon.

reflected inertia, and the torque response of the different actuation schemes. Conclusions based on this model are discussed in Section 3.1.1.

The model consists of three inertias, J_m, J_g, J_l, corresponding to the motor, gearing, and linkage, respectively, a gear ratio N with efficiency η, and a spring stiffness k (which represents either the series-elastic element or the strain-gauge). The coordinates θ_m, θ_g, and θ_l represent the angular position of the output of the motor, gearbox, and linkage, respectively. Finer details like modeling gear backlash are ignored for now to avoid unnecessary complexity. The equations of motion for this system are,

$$\left(N^2 J_m + J_g\right)\ddot{\theta}_g + k(\theta_g - \theta_l) = \eta N \tau_m, \tag{3.1}$$

$$J_l\ddot{\theta}_l + k\left(\theta_l - \theta_g\right) = \tau_l. \tag{3.2}$$

The term $\left(N^2 J_m + J_g\right)$ in Eq. (3.1) indicates that the *reflected inertia* of the motor's rotor inertia after the gearbox is scaled by N^2. Even if the inertia of the motor's rotor is small, with a high gear ratio the reflected inertia tends to be quite large. On the other hand, with a low gear ratio a larger motor may be required to achieve the desired force output, making the rotor inertia itself larger.

So how does the motor inertia scale with torque? There is some disagreement on this

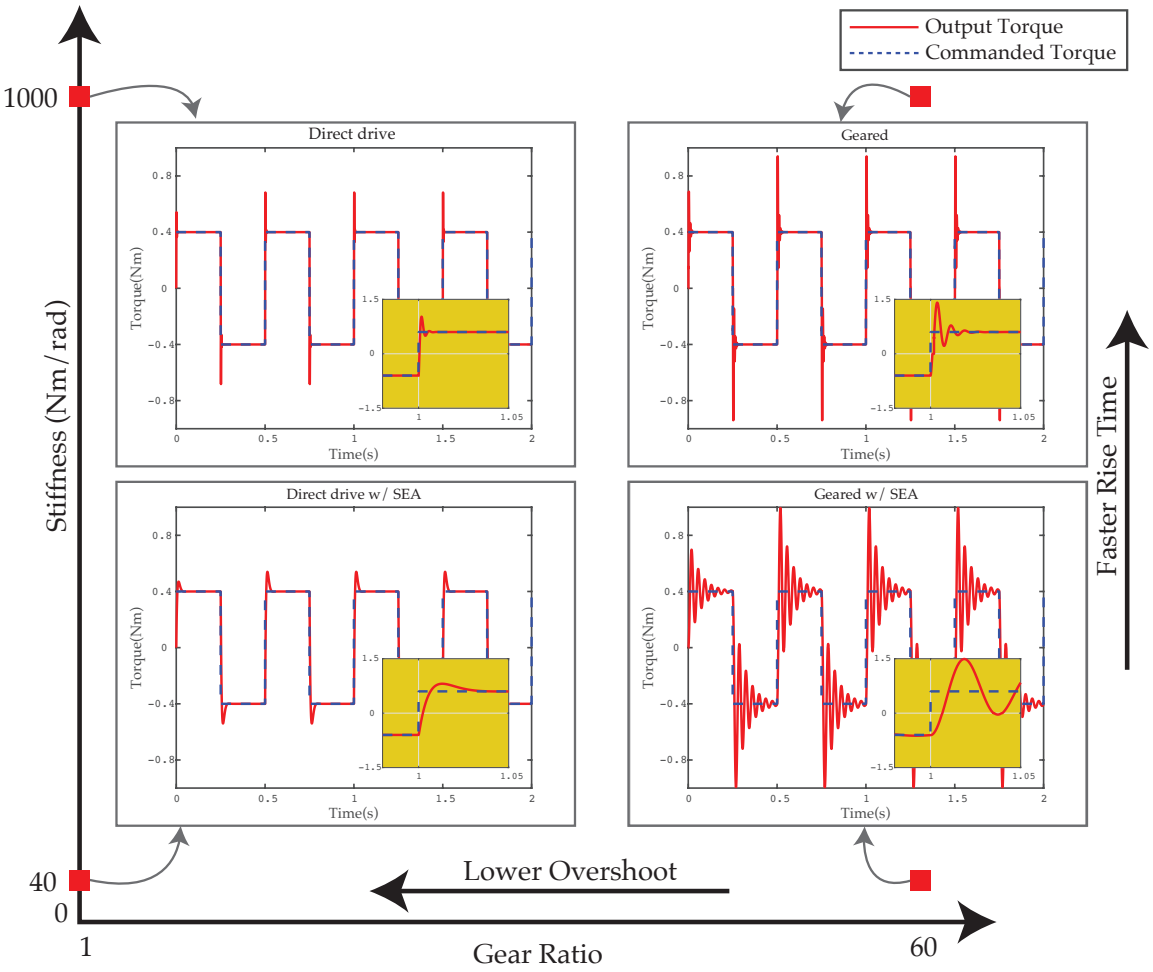


Figure 3.3: Open-loop torque response of a simulated actuator with a locked output to a square wave. The yellow inset plots are zoomed in to show the response to a single step. In particular, note the faster rise time as stiffness is increased and the reduced overshoot as gear ratio is reduced. The force bandwidth, the speed at which force can be sensed and servoed, is highest with the direct-drive actuation scheme.

point. For a fixed motor, adding a gear ratio increases the torque by N but the inertia by N^2. To get the same increase in torque by instead increasing the size of the motor, the inertia must increase by a factor between N and N^2 depending on what is held constant [63, 114, 115, 118]. Empirically, the scaling law that is seen in data from motor manufacturers, as shown in Fig. 3.2, lies somewhere in the middle. This is discussed in further detail in Section 3.4.1. Whichever model is considered, the effect is still favorable when compared to the N^2 scaling that gearboxes impose, especially when the added inertia and reduced efficiency from gearbox itself is considered. That is, to achieve the same output torque, choosing a larger motor results in a lower reflected inertia than adding a gearbox.

3.1.1 Implications of Direct-drive

Based on this simple model of actuator transmission, there are several important implications of direct-drive actuation for hand design: *transparency*, *force bandwidth*, and *speed*. This is in addition to the *mechanical simplicity* that comes from eliminating the gearbox.

A low reflected inertia indicates that the actuator is more *transparent*. That is, when a finger impacts something the loss in energy due to the motor decelerating is not as high. Equivalently, the impact does not impart as large an impulse on whatever the finger has impacted, allowing for a lighter touch.

With lower reflected inertia the finger can accelerate and decelerate faster, allowing for higher bandwidth force, velocity, or position control. Similarly, thinking of the actuator as a sensor, the input bandwidth is also higher as the world can more easily accelerate the lower inertia. Thus, the robot can feel what it is touching faster, and without the compounding issues of low-efficiency and backlash in the transmission. In some cases, the gearbox may not be back-drivable at all, precluding any sensing of the world at the motor.

The notion of *force bandwidth* is important for manipulation, and warrants a more precise definition. Reaction time of the hand to external disturbances is determined by the speed with which a force can be sensed and servoed. The choice of stiffness in the mechanism affects this bandwidth. The transfer function from motor torque to load acceleration (assuming zero load torque) is identical to the transfer function from the load torque to motor acceleration (assuming no applied motor torque). It is given by,

$$\frac{\ddot{\theta}_l}{\tau_m} = \frac{\ddot{\theta}_m}{\tau_l} = \frac{\eta N k}{(J_m N^2 + J_g) J_l s^2 + k(J_m N^2 + J_g + J_l)}. \tag{3.3}$$

This system resembles a mass-spring system $m_e \ddot{x} + k_e x = F$ with effective mass $m_e = (J_m N^2 + J_g) J_l$ and effective stiffness $k_e = k(J_m N^2 + J_g + J_l)$. The transfer function for this system is given by $1/(m_e s^2 + k_e)$. The natural open-loop frequency ω is given by

$$\omega = \frac{1}{2\pi}\sqrt{\frac{k_e}{m_e}} = \frac{1}{2\pi}\sqrt{\frac{k(J_m N^2 + J_g + J_l)}{(J_m N^2 + J_g) J_l}} \tag{3.4}$$

$$= \frac{1}{2\pi}\sqrt{k\left(\frac{1}{J_m N^2 + J_g} + \frac{1}{J_l}\right)} \tag{3.5}$$

The natural frequency is proportional to $\sqrt{k}$ (where k is the spring stiffness); so the higher the stiffness, the higher the bandwidth. Additionally, reducing the reflected and gearbox inertias $(J_m N^2 + J_g)$ also increases the bandwidth.

To evaluate the improvement in force bandwidth, in Fig. 3.3 we show simulation results of a higher-fidelity model. This model includes the effects of backlash and structural damping. The geared simulation is based on the Maxon EC-20 flat motor (serial number 241916) and a N=60:1 gearbox. For the direct-drive simulations, we scale the torque of the motor

by N and the inertia of the motor by $N^{1.5}$ to keep the torque output of both configurations comparable. An experimental comparison of these actuator schemes can be found in [56].

With this model we test the effect of stiffness and gear ratio on the open-loop torque response of the actuator to a square wave in a locked output state. Moving from unit gearing to a ratio of 60, the overshoot increases due to the increased reflected inertia. Increasing stiffness from $40\,\mathrm{N\,m\,rad^{-1}}$ to $1000\,\mathrm{N\,m\,rad^{-1}}$ improves the rise time of the system due to the reduced delay required to compress the spring.

Higher speed is another byproduct of direct-drive actuation which is well motivated by industry's need for faster cycle times. As the gear ratio moves towards unity, it unlocks access to higher speed ranges. Even if the motors are designed to work at a fixed speed, a direct-drive architecture allows the finger to move much faster than in a geared setting. Higher bandwidth, in addition to higher top speed, means that the hand can more quickly change the applied torque. This higher speed is, of course, a direct trade-off with peak *stall torque* for a given motor with different gearing.

Finally, the issue of *mechanical simplicity* is not addressed by this simple model. Beyond the additional mass and losses due to inefficiencies, gearboxes and springs tend to involve multiple moving parts that wear, deform, and can break. Eliminating these components simplifies the design and eliminates possible points of failure. Gearboxes do not add any power but take up space and add mass, and so the overall actuator power per unit volume or mass is improved by eliminating them (note that torque density may be higher or lower as a larger motor is needed.)

3.2 Metrics for Transparency

There are a few metrics that are related closely to the concept of transparency. We cover these metrics in relation to the general manipulator equation with n degrees-of-freedom and m constraints,

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{N}(\boldsymbol{q}) - \boldsymbol{J_c}(\boldsymbol{q})^T\boldsymbol{\lambda}. \tag{3.6}$$

Here, the generalized forces are $\boldsymbol{\tau} \in \mathbb{R}^n$ and the generalized coordinates are $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}} \in \mathbb{R}^n$. The mass matrix, coriolis matrix, and gravity matrix are given by $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$, and $\boldsymbol{N}(\boldsymbol{q}) \in \mathbb{R}^n$, respectively. Constraints are represented by the constraint Jacobian $\boldsymbol{J_c}(\boldsymbol{q}) \in \mathbb{R}^{m\times n}$ and constraint forces are represented by $\boldsymbol{\lambda} \in \mathbb{R}^m$.

Generalzied Inertia Ellipsoids

The Generalized Inertia Ellipsoid [4] ($\boldsymbol{u}^T\boldsymbol{\Lambda_0}\boldsymbol{u}$ for unit direction $\boldsymbol{u}$) expresses the resistance to changing the velocity of the end-effector in various direction for a human operator who holds the end-effector and applies a force with a fixed magnitude,

$$\boldsymbol{\Lambda_0} = \boldsymbol{J_c}(\boldsymbol{q})^{-T}\boldsymbol{M}(\boldsymbol{q})\boldsymbol{J_c}(\boldsymbol{q})^{-1}. \tag{3.7}$$

The Ellipsoid of Gyration [43], $\boldsymbol{u}^T\boldsymbol{\Lambda_0}^{-1}\boldsymbol{u}$ can be substituted for this metric when the Jacobian $\boldsymbol{J_c(q)}$ is not invertible.

Dynamic Manipulability Ellipsoid

The Dynamic Manipulability Ellipsoid [128, 135], $\boldsymbol{u}^T\boldsymbol{\Lambda_d}\boldsymbol{u}$ expresses the ease of changing the velocity of the end effector via the set of actuators that drive the manipulator joints by applying joint torques with a fixed magnitude,

$$\boldsymbol{\Lambda_d} = \boldsymbol{J_c(q)}(\boldsymbol{M(q)}^T\boldsymbol{M(q)})^{-1}\boldsymbol{J_c(q)}^T. \tag{3.8}$$

Effective Mass Ellipsoid

The Effective Mass Ellipsoid [65], m_e computes the effective mass along a direction $\boldsymbol{u}$ in a collision at the end-effector.

$$m_e = (\boldsymbol{u}^T\boldsymbol{\Lambda_0}^{-1}\boldsymbol{u})^{-1}. \tag{3.9}$$

Impact Mitigation Factor

[130] propose the impact mitigation factor as a metric for the contribution of reflected actuator inertias to impacts applied to the system. The impact mitigation factor is based on the difference in impulses between the system (ρ) and when its joints are locked (ρ_l,) that is, infinite reflected inertias. The generalized inertia matrix for the free and locked systems is given by $\boldsymbol{\Lambda}$ and $\boldsymbol{\Lambda_l}$ respectively where,

$$\boldsymbol{\Lambda} = \boldsymbol{J_c(q)}^{-T}\boldsymbol{M(q)}\boldsymbol{J_c(q)}^{-1}, \tag{3.10}$$

$$\boldsymbol{\Lambda_l} = \boldsymbol{J_c(q)}^{-T}\boldsymbol{M_{bb}}\boldsymbol{J_c(q)}^{-1}, \tag{3.11}$$

.

Here $\boldsymbol{M_{bb}}$ is the inertia of the COM of the locked system. $\boldsymbol{M}$ can be partitioned into block matrices to separate out the locked dynamics from the internal joint dynamics as

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M_{bb}} & \boldsymbol{M_{bj}} \\ \boldsymbol{M_{jb}} & \boldsymbol{M_{jj}} \end{bmatrix}. \tag{3.12}$$

Knowing $\rho = \boldsymbol{\Lambda v}$ and solving for equal impact velocities in both systems, the impulse in the free system can be written in terms of the impulse in the locked system as

$$\rho = \boldsymbol{\Lambda}\boldsymbol{\Lambda_l}^{-1}\rho_l. \tag{3.13}$$

Taking the difference of the impulses,

$$\rho_l - \rho = (\boldsymbol{I} - \boldsymbol{\Lambda}\boldsymbol{\Lambda_l}^{-1})\rho_l. \tag{3.14}$$

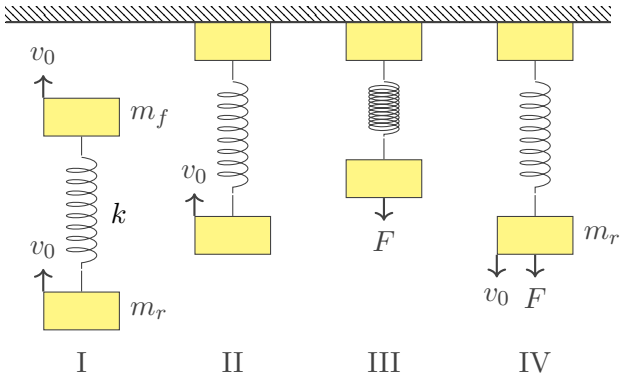


Figure 3.4: The three phases of impact: plastic impact phase, (I→II), the sensing phase, (II→III) and the reaction phase, (III→IV).

which gives us $(\boldsymbol{I} - \boldsymbol{\Lambda}\boldsymbol{\Lambda}_{\boldsymbol{l}}^{-1})$ as the impact mitigation matrix and its determinant defines the impact mitigation factor. An IMF of 1, corresponds to a system with perfect inertial backdrivability and tends towards the locked system when IMF approaches zero.

Feel-cage task

[64] test the transparency of actuators with varying transmissions using a task designed to measure how the actuator feels the environment. The task involves contact detection and caging of balls with varying mass while fixing the impact velocity. Actuators with highly geared transmissions impart more energy to the balls resulting in failed caging.

Although the feel-cage task does capture the effect of structure, sensing, and control on the transparency of a system, it has one shortcoming: the metric is binary – either the ball is caged by the actuator or it isn't. To extend this to a more quantifiable metric, we look at the total impulse transaction in a collision as the collision reflex of a system.

3.3 Proposed Collision Reflex Metric

We define the collision reflex of a robot on its performance when it detects, decelerates and moves away from an unexpected collision. The more transparent the robot, the better it's performance in the collision reflex task. The impulse in the collision is used as a metric to determine the quality of the collision reflex of the robot. The classical definition of impulse is used – the integral of force over a time interval t for which it acts.

3.3.1 1D Model of Collision

First, we analyze the impulse in a one-dimensional collision. For this collision model, we start with a model similar to the general actuator described in Section 3.2. The collision model consists of a finger mass m_f and a robot mass m_r linked together by a spring of

stiffness k. This spring lumps together the mechanical stiffness k_m (from the structure and any physical springs present), and the software-defined springs k_s in the controller. The difference between these two springs is that the rest length and stiffness of the software-defined spring is variable. The unsprung structure, transmission, and rotor inertias of the robot are lumped into the robot mass m_r while all the sprung mass is lumped into m_f. The system approaches a constraint at a pre-impact velocity of v_0 and is capable of applying a force F_a to the robot mass which causes an acceleration of $a = F_a/m_r$.

We model the robot's reflex to an unexpected collision over four events as shown in Fig. 3.4:

I. The 1D finger-spring-robot system approaches a constraint with a pre-impact velocity of v_0.

II. At time $t_0 = 0$, m_f loses all energy to the constraint in a plastic impact. As m_r is still approaching the constraint with velocity v_0, it starts compressing the spring. We assume the robot is velocity controlled for this stage.

III. When a collision event is detected at time t_1, the robot switches to a force controller resulting in an acceleration a of the robot mass away from the constraint.

IV. The collision event completes at time t_2 when the spring is not applying any more force on the constraint.

3.3.2 Phases of Impact

We analyze the collision in 3 phases: plastic impact phase, $I_{I\to II}$, the sensing phase, $I_{II\to III}$ and the reaction phase, $I_{III\to IV}$. The collision reflex impulse is then the sum of the impulse contribution from each phase:

$$I = I_{I\to II} + I_{II\to III} + I_{III\to IV}. \tag{3.15}$$

Plastic Impact Phase

In this phase, the finger mass collides with the constraint and dissipates all of the momentum it carries. The impulse applied in this phase is thus the change in momentum of the finger as it comes to a instantaneous stop:

$$I_{I\to II} = m_f v_0. \tag{3.16}$$

Sensing Phase

At the end of the plastic impact phase, system still has no information about the contact event and proceeds as usual. In this analysis we assume that the usual behavior of the

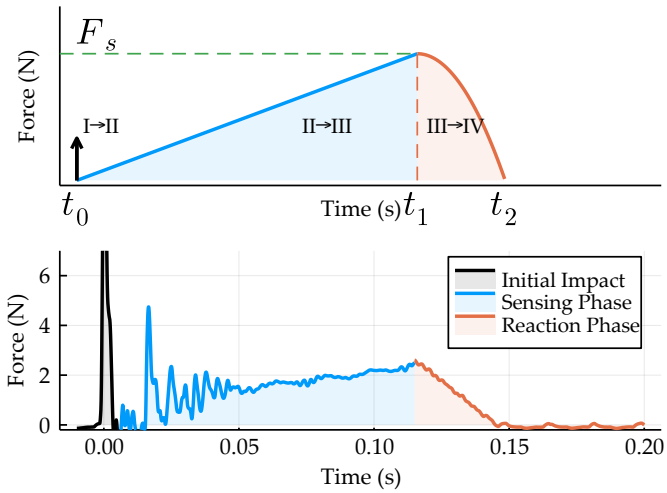


Figure 3.5: The three phases of impact are shown for the model as a force-time plot with significant features and time-points labeled (top.) Experimentally collected force-time profile of a collision for a U8 motor (bottom.) Details of the experiment can be found in Section 5.2.

system is to track a velocity reference for the robot mass. That is, until a collision is sensed, the robot mass is attempting to maintain a constant velocity v.

This velocity controller presents as a ramp signal in the force-time signature of the collision up to the point of contact detection at time t_1. In this model the point of contact detection is represented by a threshold F_s.

To compute the impulse $I_{II\to III}$ which is contributed during this stage, we first solve for the collision sensing time t_1:

$$t_1 = \frac{F_s}{kv_0}. \tag{3.17}$$

The impulse in this phase is the area of the triangle shaded blue in Fig. 3.5,

$$I_{II\to III} = \frac{1}{2}F_s t_1 = \frac{F_s^2}{2kv_0} \tag{3.18}$$

Reaction Phase

Once contact is detected, the robot is commanded an acceleration a away from the constraint. If the robot mass m_r is not at rest at the end of the sensing phase, part of the reaction phase would be used to decelerate the robot mass. In our analysis, we assume the contribution of this phase to be zero i.e. the reaction phase starts with the robot mass at zero velocity. The stiffness of this model has a discrete change to include only the mechanical stiffness of the robot and any software-defined stiffnesses or gains are instantaneously zeroed. Note that up until now there were two sources of stiffness: one from the mechanical elements (flex in the structure, strain gauges, fingertips/skin, etc) and one from software

elements like control gains. The stiffness from the mechanical elements is fixed while the software stiffness is variable. We assume that elasticity from software with variable stiffness and rest length can be instantaneously zeroed. Thus, the only stiffness that needs to be overcome by the reaction phase is then the mechanical stiffness.

The acceleration command is assumed to start promptly after detection of collision at t_1 and ends at t_2, the time at which the spring has returned to its resting length. To compute the impulse $I_{III\to IV}$ for this phase, we first need to compute t_2. This can be achieved by studying the motion of the robot mass starting at a displacement F_s/k_m with zero velocity and ending at zero displacement. Using the laws of motion $s = ut + \frac{1}{2}at^2$ where s is the displacement in the mechanical compliance, u is the initial velocity (which is zero in this case,) a the acceleration, and t is the time of motion. Solving for the time of motion, we get $t_2 = t_1 + \sqrt{\frac{2F_s}{ak_m}}$. The force during the reaction phase follows the profile defined by $F_2 = F_s - k_m a t^2/2$ Integrating F_2 from 0 to $t_2 - t_1$, we get the impulse during the reaction phase:

$$I_{III\to IV} = F_s(t_2 - t_1) - \frac{1}{6}k_m a(t_2 - t_1)^3 = \sqrt{\frac{8F_s^3}{9ak_m}}. \tag{3.19}$$

Cases where the contribution of the deceleration of the robot mass plays a prominent role in the collision reflex are easy to account for. The initial velocity u is no longer zero and is informed by the leftover momentum of the robot mass after the sensing phase. This would lead to changes in the expressions for u, t_2 and I_2 leading to added complexity in the collision reflex. As we will see in our experimental validation (Section 5.2), the reaction phase does not contribute prominently to the collision reflex. We can therefore safely assume that the robot mass is at rest at the end of the sensing phase.

3.3.3 Total Impulse in a Collision

After solving the contribution of each phase to the total impulse, we can add Eq. (3.16), (3.18), and (3.19) to determine the total impulse I.

Definition 1 (Collision Reflex Metric)**.**

$$I(m_f, F_s, k, k_m, v_0, a) = m_f v_0 + \frac{F_s^2}{2kv_0} + \sqrt{\frac{8F_s^3}{9ak_m}} \tag{3.20}$$

The collision reflex metric depends on the finger mass m_f, sensing force F_s, stiffness k, pre-impact velocity v_0 and maximum allowable acceleration a.

The collision reflex assumes that the dynamics of the initial impact are much faster than the sensing bandwidth. If this is not true, and the sensor can trigger a collision detection event on the initial impulse, the sensing phase gets absorbed into the initial impulse. The robot mass at the end of the sensing phase can no more be assumed to be at rest.

3.3.4 Pre-impact Velocity for Minimum Impulse

The analytical expression for the impulse in a collision for a generalized actuator shows dependence of velocity for the impact and sensing phases. The total impulse I in Eq. (3.20) shows that the reaction phase impulse $I_{III \to IV}$ does not depend on the pre-impact velocity of the system. Differentiating I twice with respect to v_0,

$$\frac{\partial^2 I}{\partial v_0^2} = \frac{F_s^2}{k v_0^3}. \tag{3.21}$$

As F_s^2 and k cannot be negative, $\partial^2 I / \partial^2 v_0 > 0$ for all positive pre-impact velocities and the function I is convex and has a single minima. The optimal velocity for minimal impulse ($\partial I / \partial v_0 = 0$) is

$$v_0^* = \frac{F_s}{\sqrt{2 k m_f}}. \tag{3.22}$$

The minimum impulse at the optimal impact velocity v^* is

$$I^* = \frac{F_s \sqrt{2}}{\omega_s} + \frac{F_s}{\omega_a} \sqrt{\frac{8 F_s}{9 F_a}}, \tag{3.23}$$

where $\omega_s = \sqrt{k/m_f}$, the open-loop sensing bandwidth (natural frequency) and $\omega_a = \sqrt{k/m_r}$ the open-loop actuation bandwidth of the system.

This analysis highlights relationships between the properties and behavior of the system during collision. Stiffness and mass are infinite in an ideal rigid body. As the stiffness of the system increases, the system behavior shifts towards an ideal rigid body collision, i.e. minimum impulse is at zero pre-impact velocity. We also see an inverse relationship between the open-loop bandwidth ω_a of the finger-mass and spring system and the impulse. As the open-loop sensing bandwidth goes up, the impulse reduces. The reaction phase is dependent on the open-loop actuation bandwidth of the system ω_a. As the open-loop actuation bandwidth increases, the acceleration capability of the robot improves reducing the impulse applied during the reaction phase.

The collision reflex allows for a quantitative measure of the transparency of the system which is only dependent on system parameters like pre-impact velocity, inertia, acceleration, and stiffness. Using this tool, we can directly compare the performance of two systems.

3.4 Actuator Selection and the Collision Reflex

With the ability to compare transparency of systems using the collision reflex metric, we can start to understand the tradeoffs between actuation schemes. Here we consider a space of actuation schemes spanned by two axes: stiffness, and reflected inertia. There are 4 quadrants of this space: high stiffness, high reflected-inertia (e.g. gearmotor + strain-gauge); low stiffness, high reflected-inertia (e.g. gearmotor + series-elastic actuation [99]); high

Scaling Law	Mass	Reflected Inertia	Torque
Motors			
Isometric [118]	lr^2	lr^4N^2	lr^4
Empirical† [26]	lr^2	lr^4N^2	$lr^{2.8}$
Quadruped design [63, 114]	lr	lr^3N^2	lr^2
w/ electrical & thermal [37]	lr^2	lr^4N^2	$lr^{2.5}$
Gearboxes [106]			
Parallel Shaft	lr^2	$lr^4N^2a^{-1}$	lr^2a^{-1}
Planetary	lr^2	$lr^4N^2a^{-1}$	lr^2a^{-1}
Harmonic Drive	lr^2	lr^4N^2	r^3
Cycloidal Drive	lr^2	lr^4N^2	r^4l^{-1}
Ball Screw	lr^2	lr^4	r^3

Table 3.1: Scaling Laws for mass, inertia and torque of motors and gearboxes. Here l, r are the length and radius of the motor/gearbox, N is the transmission ratio of the gearbox with a stages. † assumes inertia and mass follow isometric scaling.

stiffness, low reflected-inertia (e.g. direct-drive [5]); low stiffness, low reflected-inertia (e.g. compliant direct-drive [10, 30]). In this section, we look at actuation models to understand the effect on the collision impulse with changes in the reflected inertia and torque output of an actuator with one degree of freedom.

3.4.1 Actuator Scaling

Torque and inertia of an actuator are closely related. Choosing a bigger motor for higher torque output will also increase its rotor inertia. A smaller motor on the other hand will produce insufficient torque for the application requiring a transmission. Reflected inertia of a gear motor scales the rotor inertia by a factor of N^2.

Various empirical and analytical models show that the scaling between rotor inertia J_m and torque τ_m lies somewhere between linear and quadratic depending on assumptions. If isometric scaling is assumed [118], i.e. the length of the motor scales with the radius, $J_m \propto \tau_m$. Empirical analysis of motors [26] show $J_m \propto \tau_m^{1.32}$ assuming inertia and mass follow isometric scaling. Quadruped design studies assume fixed radial thickness and motor length [63, 114] to arrive at $J_m \propto \tau_m^{1.5}$, or assume fixed radial thickness and mass [115] to arrive at $J_m \propto \tau_m^2$. Lastly, deriving the scaling laws based on electrical and thermal dynamics [37] results in $J_m \propto \tau_m^{1.6}$.

When smaller motors are employed, a transmission is added to achieve a torque requirement. The contribution of reflected inertia by these transmissions may not be negligible, e.g. [107] shows the scaling laws for parallel shaft and planetary gear trains, harmonic and cycloidal drives and ball screws. We ignore the effect of this contribution to simplify the analysis and consider only the reflected rotor inertia.

Parameter	Unit	M1	**M2**	M3
Stator OD (r_m)	mm	10.0	**60.0**	100.0
Stator ID	mm	6.95	**41.7**	69.5
Rotor OD	mm	4.83	**40.6**	48.30
Rotor ID	mm	3.36	**31.0**	33.57
Stack Height (l)	mm	12.5	**12.5**	12.5
Rotor Inertia (J_m)	$\mathrm{kg\,m^2}$	1.71E−8	**2.21E−5**	1.71E−4
Cont. Torque (τ_c)	N m	5.94E−3	**0.524**	1.879
Peak Torque (τ_m)	N m	1.47E−2	**1.3**	4.661
Gear Ratio (N)	-	88.18	**1**	1
Inertia ($N^2 J_m$)	$\mathrm{kg\,m^2}$	1.33E−4	**2.21E−5**	1.71E−4
Peak Torque Out ($N\tau_p$)	N m	1.3	**1.3**	4.661

Table 3.2: Motor parameters of the Celera Motion Omni+ motor (OPN-060-013-A, M2 in the table) used for analyzing the effect of actuation selection on the collision reflex. The scaled parameters for motors with stator OD of 10 mm (M1) and 100 mm (M3) are also shown.

The scaling laws for motors and gearboxes are summarized in Table 3.1.

3.4.2 Effect of Actuator Selection

How does the selection of an actuation scheme affect the collision reflex of a system? To understand the relationship, we select a direct-drive motor and scale the motor radius keeping the length and the minimum torque output constant. A transmission is added to overcome the lower torque capability of smaller motors. As scaling laws from [37] shows better correlation to empirical data [107] than the Quadruped laws [63, 114], they are applied to determine the inertia and torque output of the actuator. We assume that the effects of added inertia from the transmission are ignored and that all generated actuator configurations are feasible. The parameters of the selected motor (Celera Motion Omni+ series 60 mm frameless motor OPN-060-013-A) and two scaled motors are shown in Table 3.2.

Fig. 3.6 shows the variation in collision reflex with changes in velocity and radius of the actuator. Three operating ranges of motor scaling are seen. For small radius motors with high gear ratios, the impulse increases dramatically as the reflected inertia is scaled by the square of transmission ratio N. On the other end of the scale, large radius motors scale the impulse as the reflected inertia scales with the scaling law $J_m \propto r^{2.5}$. In the middle a region with low impulse is found which represents the quasi-direct-drive or direct-drive regime (low or unity transmission ratio) designs. Note that without the inclusion of reflected inertia added by the gearbox, optimal motor scaling lies somewhere in this range.

From Eq. (3.20), the dependence of the impulse on velocity follows $I(v) = av + b/v + c$ where a, b and c are contributions of the three phases at unit pre-impact velocity, respectively. Lower velocities see the contribution of phase two dwarfs the other phases due to

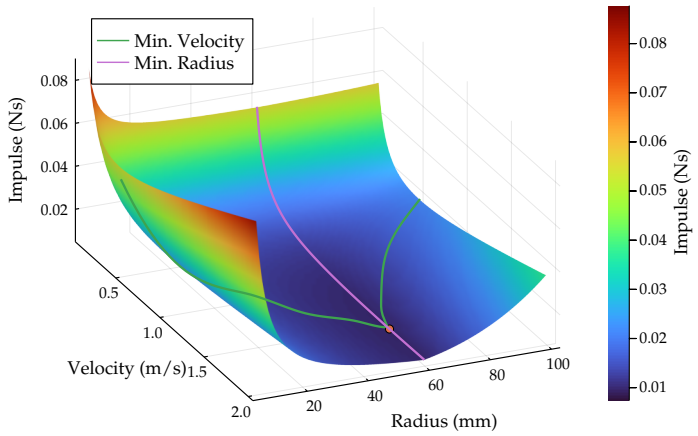


Figure 3.6: Variation of the collision impulse as velocity and radius are varied with 1000 N/m mechanical stiffness (right). The software stiffness k_s was set at 100 N/m with a collision sensing threshold F_s of 3 N. The orange dot on the plot shows the optimal parameters for the minimum impulse.

the slow ramp up to F_s. At higher velocities, phase one dominates when the inertia of the system is high, i.e. when the gear ratio is high or when larger than required motors are selected. These trends can be seen in Fig. 3.6, with the optimal velocity lying somewhere in the middle.

Fig. 3.7 shows the variation in collision reflex with changes in mechanical stiffness and sensing strategy. As the mechanical stiffness increases, the contribution of phases two and three decrease, Eq. (3.18) and Eq. (3.19). At higher stiffness, the impulse reduces assuming the sensing force threshold F_s remains constant. If the collision detection is implemented as a threshold on error in position control instead, the sensing threshold is proportional to the stiffness. This will cause the impulse to be higher as stiffness increases (as seen in Section 5.2.)

3.5 Collision Reflex in Higher Dimensions

The collision reflex metric is also useful in design and planning tasks for higher dimensional systems.

To this end, we extend the general manipulator described in Section 3.2. If we apply the actuator model discussed in Section 3.2 at each joint of the manipulator, we can decouple the actuator (j) dynamics from the structure (b). The new manipulator dynamics have a mass matrix as

$$\tilde{\boldsymbol{M}}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{M_{bb}} & \boldsymbol{M_{jb}} \\ \boldsymbol{M_{bj}} & \boldsymbol{M_{jj}} \end{bmatrix}. \tag{3.24}$$

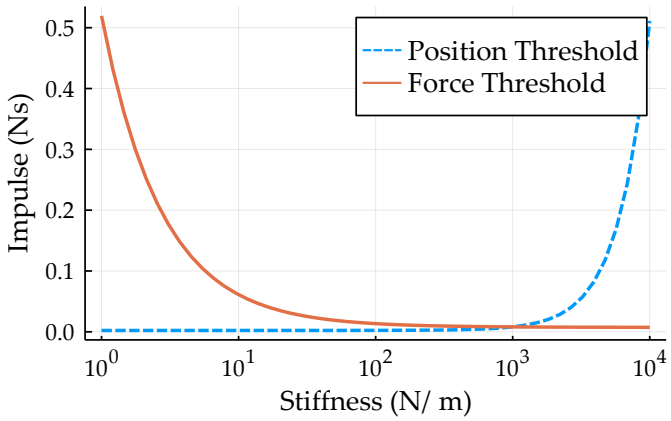


Figure 3.7: Effect of change in stiffness on the collision reflex impulse for two sensing schemes: position thresholding and force thresholding.

The dynamics of the structure and the actuators are coupled by a stiffness matrix

$$\tilde{\boldsymbol{K}} = \begin{bmatrix} \boldsymbol{K} & -\boldsymbol{K} \\ -\boldsymbol{K} & \boldsymbol{K} \end{bmatrix}, \tag{3.25}$$

where $\boldsymbol{K} = diag(k_1 \dots k_n)^T$ is a diagonal stiffness matrix with its diagonal elements describing the stiffness at each joint.

Let us look at the impulse transferred in a collision at the contact frame c along a direction u, e.g. as in Fig. 3.8. We can apply the one dimensional collision reflex metric, Section 3.3, in this direction by using the directional effective mass, force, stiffness, and velocity in task space coordinates. The initial impulse $I_{II \to III}$ is caused by the inertia $\boldsymbol{M_{bb}}$ which describes the structure of the robot. The impulse during phases two ($I_{III \to IV}$) and three (I_3) are caused by the spring stiffness, $\boldsymbol{K}$ and the inertia of the actuators, $\boldsymbol{M_{jj}}$.

We can transform the inertias $\boldsymbol{M_{bb}}$ and $\boldsymbol{M_{jj}}$ and stiffness $\boldsymbol{K}$ defined in the generalized coordinate to task space coordinates through the Jacobian $\boldsymbol{J_c(q)}$. We can then map to the equivalent one-dimensional representation by projecting on the collision normal $\boldsymbol{u}$. After applying Eq. (3.9), the effective finger mass m_f, robot mass m_r, and stiffness k at the contact point $\boldsymbol{c}$ in a collision direction $\boldsymbol{u}$ are

$$m_f = (\boldsymbol{u}^T(\boldsymbol{\Lambda})^{-1}\boldsymbol{u})^{-1}, \tag{3.26}$$

$$m_r = (\boldsymbol{u}^T(\boldsymbol{\Lambda_a})^{-1}\boldsymbol{u})^{-1}, \tag{3.27}$$

$$k = \boldsymbol{u}^T\boldsymbol{K_t}\boldsymbol{u}. \tag{3.28}$$

where, $\boldsymbol{\Lambda} = \boldsymbol{J_c}^{-T}\boldsymbol{M_{bb}}\boldsymbol{J_c}^{-1}$ and $\boldsymbol{\Lambda_a} = \boldsymbol{J_c}^{-T}\boldsymbol{M_{jj}}\boldsymbol{J_c}^{-1}$ The task space stiffness matrix $\boldsymbol{K_t}$ can

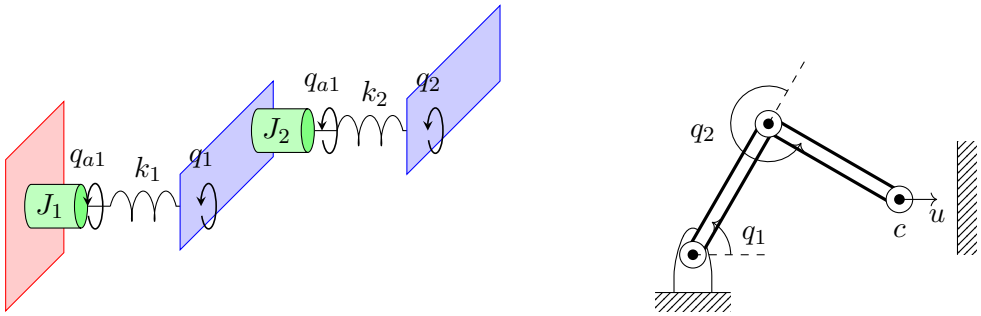


Figure 3.8: Schematic of a two-link manipulator colliding with a constraint. The quantities show the collision in the task space.

be computed through another Jacobian map as,

$$\boldsymbol{\tau} = \boldsymbol{K}\Delta\boldsymbol{q}, \tag{3.29}$$

$$\boldsymbol{J}_{\boldsymbol{c}}^{T}\boldsymbol{f} = \boldsymbol{K}\boldsymbol{J}_{\boldsymbol{c}}^{-1}\Delta\boldsymbol{x}, \tag{3.30}$$

$$\boldsymbol{f} = (\boldsymbol{J}_{\boldsymbol{c}}^{-T}\boldsymbol{K}\boldsymbol{J}_{\boldsymbol{c}}^{-1})\Delta\boldsymbol{x}, \tag{3.31}$$

$$\boldsymbol{K}_{\boldsymbol{t}} = \boldsymbol{J}_{\boldsymbol{c}}^{-T}\boldsymbol{K}\boldsymbol{J}_{\boldsymbol{c}}^{-1}. \tag{3.32}$$

The task space acceleration capability of the robot a is given by the Jacobian transpose map.

$$a = \frac{1}{m_r}\boldsymbol{u}^T\boldsymbol{J}_{\boldsymbol{c}}^{-T}\mathrm{sat}(\boldsymbol{J}_{\boldsymbol{c}}^{T}\boldsymbol{u}). \tag{3.33}$$

where the sat$(\cdot)$ function saturates the motor torque to its maximum allowable limit.

Having identified the one-dimensional collision parameters, m_f, k, and a (Eqs. 3.26, 3.28, 3.33 respectively), impulse for any collision frame c, pre-impact velocity v_0, force sensing threshold F_s, and impact direction $\boldsymbol{u}$ can be computed.

As an example, we apply the n-dimensional collision reflex extension to a 2 degree of freedom (DOF) manipulator. The manipulator shown in Fig. 3.8 is modeled with the Celera Motion motor discussed in Section 3.4 to understand the effect of motor scaling in higher degree of freedom robots.

By sweeping vector $\boldsymbol{u}$ a full rotation about the end-effector, a collision reflex surface for a given configuration of the manipulator is generated. Fig. 3.9 shows the collision reflex surface for a 2 DOF manipulator at various configurations $(q_1, q_2)^T$, motor scales r_m, and pre-impact velocities v_0.

As q_2 approaches 0, the collision reflex surface elongates to show the infinite stiffness along the singularity. The surface also narrows in the orthogonal direction.

Motor scaling shows a trend similar to the 1D analysis (Fig. 3.6.) The surface resembles a belted ellipsoid which is characteristic of the effective mass ellipsoid from [65]. This belted ellipsoid elongates as we move away from direct-drive ($r_m = 60\,\mathrm{mm}$). The inertia orthogonal to the second link is isolated by the spring at the second joint and does one depend on the

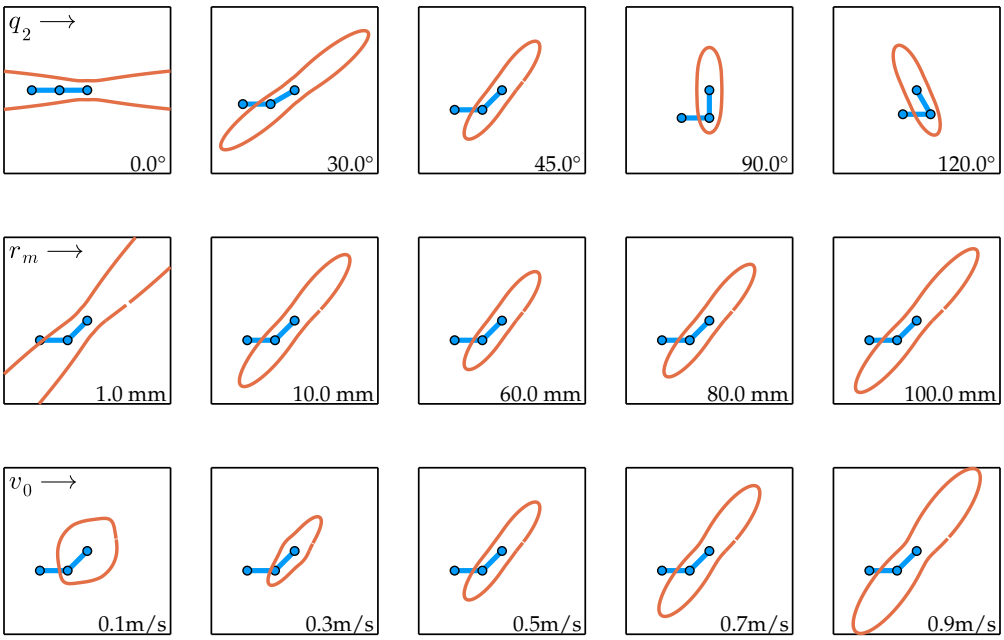


Figure 3.9: The collision reflex surface for a two link manipulator. (Top row) varying configuration; (middle row) varying motor scale; (bottom row) varying velocity. The robot is analyzed at $q_2 = 45°$, $r_m = 60.0\,\text{mm}$, and $v_0 = 0.5\,\text{m/s}$ unless otherwise indicated.

reflected inertia of the actuator. No change to the surface is seen in that direction.

At slower velocities, the collision reflex is dominated by the sensing phase where $I_{II \to III} \propto 1/v$, Eq. (3.18). As the sensing phase does not depend on the inertia of the actuator, the surface approaches a circle – isometric impulse in all directions. Increasing the velocity of impact has two effects: (1) the surface elongates perpendicular to the second link as initial impact (which depends on inertia, Eq. (3.16)) starts to dominate; (2) the overall surface shrinks up to $v_0 = v^*$ and then expands (see Eq. (3.22)).

3.6 Discussion

A robot's transparency and capability to mitigate collisions depends on structural parameters, actuators, sensing, and control architectures. In this chapter, we proposed the collision reflex metric to quantify this capability. The metric builds upon the feel-cage task and avoids the dependence on parameters of the external object to be caged. An analytical expression for the collision reflex is derived in one-dimension and extended to higher dimensions.

The three phases of collision discussed in Section 3.3 play varied roles depending on the pre-impact velocity. At high velocities, the impulse during collision behaves as one would expect – the initial impact dominates and the impulse intuitively follows the change in

momentum of the reflected inertia at the contact point. The behavior at lower speeds is rather counter intuitive, as the sensing phase dominates the impulse. As you increase the pre-impact velocity, the total impulse decreases. We show that this operating regime, where the impulse reduces as the speed increases, exists on real systems at relevant velocities. However, note that this behaviors is highly dependent on the sensing and control architecture.

Even so, no matter what the structure, sensing, and control are for a system, the metric is generalizable. The experiment to measure the collision reflex of any system, real or simulated, is simple. The system is impacted against a rigid surface and the area under the force-time curve for that event gives the collision reflex metric for that system.

The collision reflex metric considers the total impulse during a collision and not just the maximum force (as in [67]). Why not use the maximum force then? The impulse metric gives you more information about the transparency as it depends both on the sensing bandwidth an actuation bandwidth. A metric based on maximum force will be easily defeated by mechanical compliance which has a negative effect on actuation bandwidth.

Chapter 4

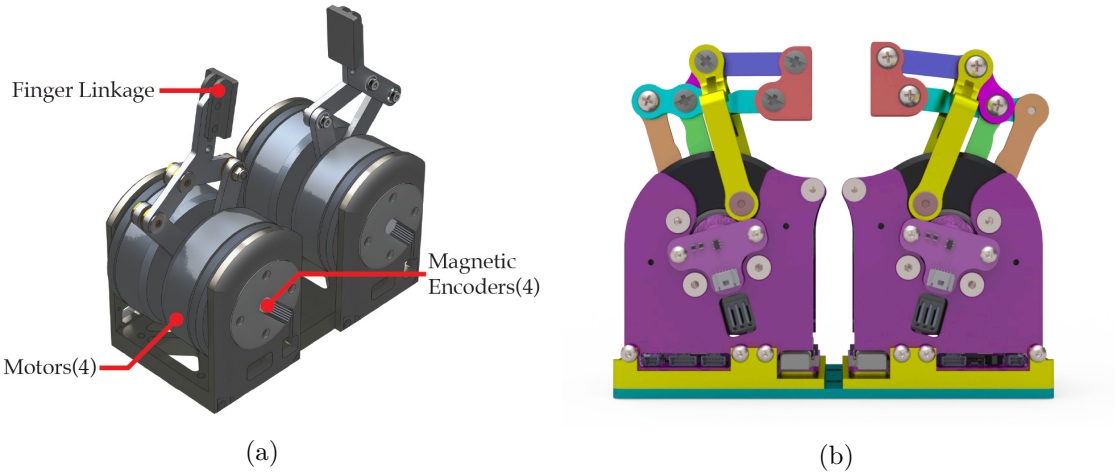
Design and Control of a Direct-drive Hand

Introduction

In an analysis of transmissions, we have shown direct-drive transmissions to be well suited for transparent end-effector design. The choice of transmissions is one of many design considerations in the specification of a direct-drive gripper. The mechanical, electronic and control choices are discussed in this chapter. Supporting infrastructure and software architecture for the combined robot-hand system and associated controllers is also covered.

Design Parameters	Value
Rated Continuous Torque (per motor)	0.3N m
Force at Fingertip	6N
Maximum Speed (tested)	200rpm
Rated Voltage	22.5V
Operating Voltage	24V
Motor Driver Continuous Current	10A
Communication Protocol	CAN @ 1kHz
Mechanical Bandwidth	$\geq$12Hz
Degrees of Freedom	4
Parallel Stroke	90mm
Weight	0.9kg
Motor Weight (per motor)	0.18kg
Encoder Resolution	4096cpr
Parts Cost	$\sim$\$1000

Table 4.1: Specifications for the DDHand



(a) (b)

Figure 4.1: (a) Design of the first prototype of the two-finger DDHand. (b) Updated linkage with simplified kinematics and fingertips constrained to be parallel.

4.1 Design of the DDHand

The design of the Direct-Drive Hand (DDHand) is shown in Fig. 4.1. The mechanical simplicity of this robot hand shows in its bill of materials. The prototype hand has a parts cost of under $1000 in single quantity. The design compares favorably with existing single degree-of-freedom industrial grippers as there is no gearbox or series spring. The smaller number of moving parts per DOF also implies higher reliability and lower wear and tear.

The design of the hand is modular: each finger is a self-contained module with its own set of drive electronics and actuators. The hand can be easily extended to a three-finger design by updating the module mounting plate and assembling 3 finger modules. A summary of specifications is shown in Table 4.1.

Mechanical Design and Actuators

The finger modules, inspired by the Minitaur [63] and other robots [18, 62], feature a parallel 5-bar linkage connected to two brushless gimbal motors. Finger modules can be arranged into parallel and spherical hand designs to support a variety of tasks. A two finger parallel configuration is presented here.

The linkage design allows the finger to squeeze with twice the force that one motor can produce and reduces the torque requirement on the motors. Placing both motors at the base of the finger ensures that the linkage inertia is low and the fingers can be accelerated or decelerated quickly.

Each finger module has two T-Motor GB54-2 brushless DC (BLDC) gimbal motors [120]. A continuous torque of 0.3N m (at 24V) per motor has been experimentally verified. With a finger length of 10cm, the DDHand can produce a force of 6N at the fingertips. We chose

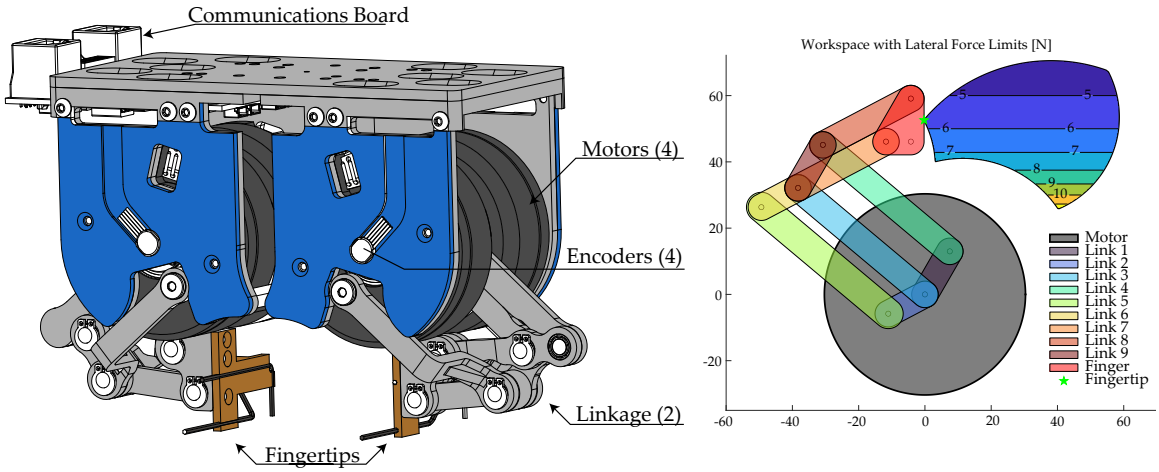


Figure 4.2: Fingertip design (left), schematic (center) and workspace (right) of the Dexterous DDHand.

gimbal motors for three reasons: (1) The are low cost and easily available due to the boom in Do-It-Yourself hobby drones; (2) gimbal motors are wound for high voltage, low current applications (producing less heat and enabling the use of smaller MOSFETs); and (3) they are designed for gimbals which are mostly operated at close to zero speed. The low speed is suitable for manipulation as it is rather unlikely that these motors will be required to spin at speeds higher than 500rpm.

The linkage plays an important role in enhancing the capability of this hand. See [6, 63] for a discussion of linkage designs. The linkage used in the first prototype shown in Fig. 4.1a, was empirically chosen to allow parallel grasps of large and small objects and some variability in the finger angle to enable pinch grasps.

For improved dexterity, we have updated the linkage design of the DDHand (Fig. 4.1b) to include:

Simplified linkage kinematics The 5-bar finger linkage has been converted to a parallelogram linkage to simplify computation of linkage forward and inverse kinematics, and the Jacobian. This allows the linkage to behave as a two-link serial linkage while avoiding the problem of moving actuator mass.

Parallel Motion We add two more parallelogram linkages to propagate a ground reference to the finger tip in order to keep the fingers parallel to each other at all points in the finger workspace.

The schematic and linkage of the *Dexterous* DDHand are shown in (Fig. 4.2). A new 9-bar linkage was designed with three parallelogram closed-chains to mimic a parallel jaw gripper. The linkage behaves like a 2-bar serial linkage simplifying the closed-form solution for computing the forward and inverse kinematics and the Jacobian. Propagating a ground reference up the chain keeps the fingertips parallel to each other. This design choice was made to reduce complexity in the primitive design phase. The configuration, workspace

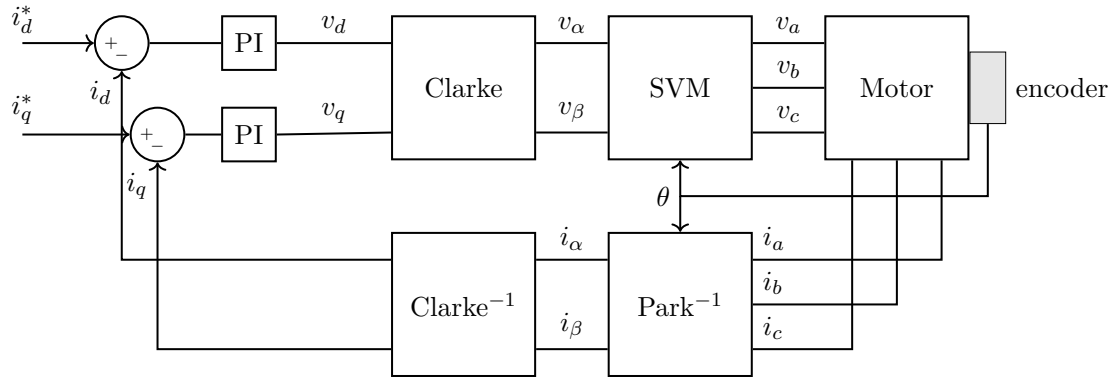


Figure 4.3: A schematic of the FOC loop for current control in the motor controllers. The field-oriented controller drives the direct current component (i_d) to zero. Torque produced is proportional to the quadrature component (i_q.) The drives sense motor position, bus voltage, and three phase current and voltage.

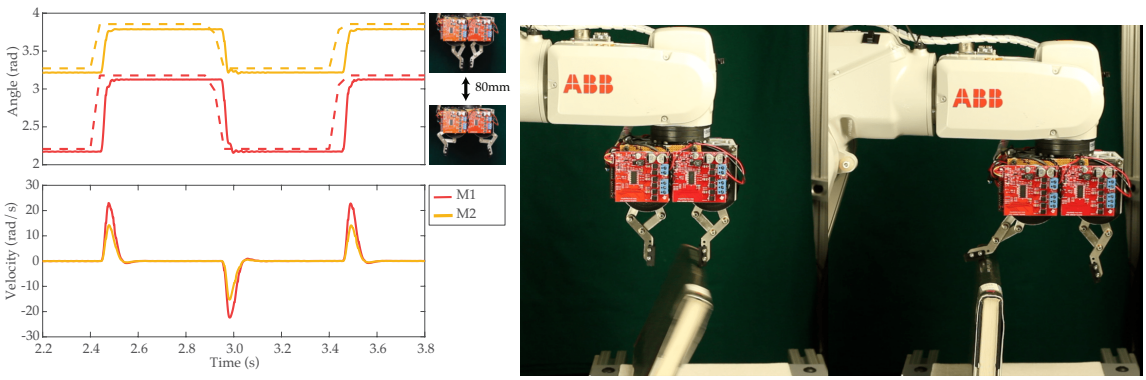
and force application capability for the linkage is shown in Fig. 4.2. The rotor and stator bolt-circles were used as links to package the linkage in the tight space.

Motor Control

Each finger module is controlled by a custom BLDC motor driver developed in the lab. The drivers are based on the InstaSPIN Field-Oriented Control (FOC) technology from Texas Instruments [121] using two F28069M launchpad and four BOOST-XL DRV8301 booster packs. Each controller module has two three-phase drivers which can output a continuous current of 10A on each driver. Position feedback is achieved with an on-axis magnetic encoder from RLS, which, along with a diametrically polarized magnet, provides 12-bit absolute position sensing. These boards are powered with 24V and communications are handled by a micro-controller on each driver via Controller Area Networking (CAN) bus at 1kHz. The CAN bus can support up to three finger modules at this frequency. The communication rate will suffer on the addition of more finger modules. In future iterations, the control electronics will be integrated into the chassis of the DDHand. A custom pcb has been designed which shrinks the motor controllers for packaging inside the finger module.

The electronics for the Dexterous DDHand were moved off-board close to the base of the robot and the motor conductors were routed through a 14-conductor wiring hardness and the encoder signals were differentially transmitted over two RJ45-terminated CAT5e cables. The rest of the system architecture is carried over from [11].

A Proportional-Derivative scheme is used for position control of each motor. The output current is tracked by a FOC loop (Fig. 4.3.)



(a) (b)

Figure 4.4: (a) A step response of one finger module of the DDHand showing the angular position and velocity of the motors. The dotted lines show the commanded positions. The delay seen between the command and response of the motor captures the round trip latency forCAN communication. The lack of an integral gain results in a small steady-state error. (b) The effect of a transparent finger on an unstable object. In the case of the stiff gripper (left: DDHand with high gains), the book topples. A transparent gripper (right: DDHand with the motors off) is able to maintain contact with the book and prevent toppling.

4.2 Experimental Validation

Step Response

To demonstrate the enhances in speed and bandwidth of the DDHand, an step input of 80mm is commanded in the parallel grasp mode to the finger modules (Fig. 4.4a). The position and velocity of both motors in one finger module is shown. The motors have a communication delay of approximately 0.05s and a rise time of 0.03s. A steady-state error is seen due to the lack of an integral gain on the motors. This choice was intentionally made to more accurately model the motors as a spring-damper system.

Due to the low reflected inertia, the DDHand can achieve high bandwidth control for dynamic manipulation. A conservative estimate for the bandwidth of the system can be obtained from the rise time [88] as,

$$B = \frac{0.35}{t_r}. \tag{4.1}$$

With this relation, the bandwidth of the hand is estimated at 12Hz. In comparison, the Schunk gripper WSG50 [113] has a stroke length of 55mm, with a maximum speed of $420\text{mm}\,\text{s}^{-1}$ and a maximum acceleration of $5000\text{mm}\,\text{s}^{-2}$. This means the Schunk gripper takes 0.17s to cover 35mm at max acceleration. Using the above relation, the bandwidth estimated for the Schunk gripper is approximately 2Hz.

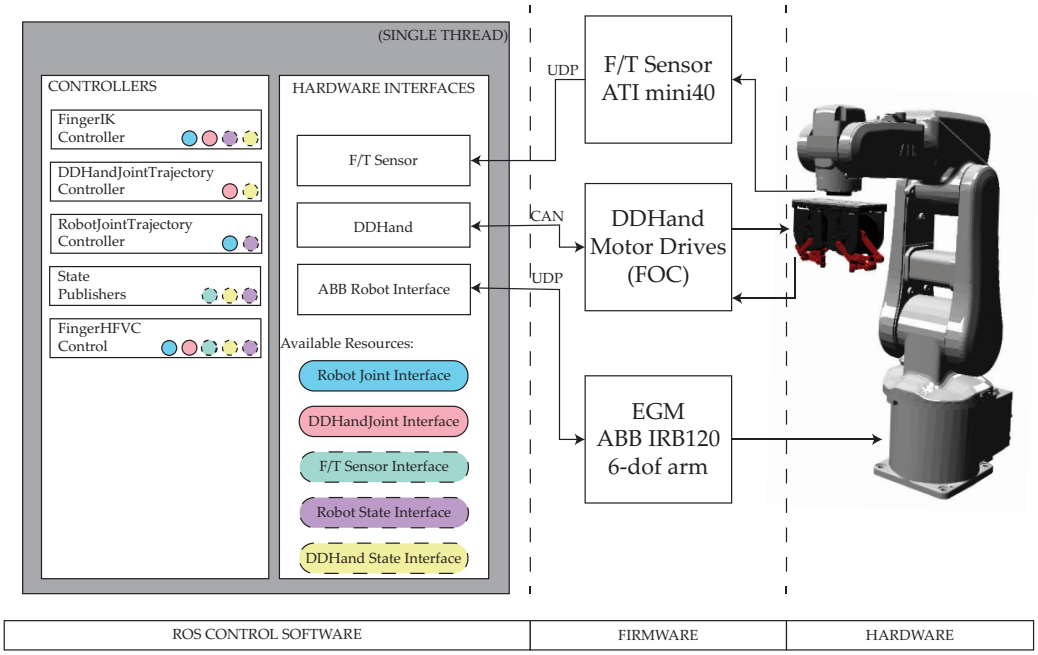


Figure 4.5: Infrastructure and software architecture of the DDHand, ABB IRB 120 robot and ATI Mini40 F/T sensor. ROS Control framework is used for all real-time control implementation providing a modular interface for switching, starting and stopping controllers. Controllers interface with hardware or simulated hardware through the same specification.

Transparency and Variable Impedance

If a robot gripper has to interact with a fragile or unstably balanced object like an empty bottle there is a danger that the object will be broken or toppled. This is because there is no way to manage the impulse imparted by the high reflected inertia of the motor rotors through the gearbox. Even if a spring is used, it is generally preferred to use a spring with higher stiffness to preserve the quality of the position controller and apply high grasping forces. In contrast, the low reflected inertia of the DDHand mitigates this impulse mechanically. Fig. 4.4b shows frames from such a contact with a book standing on its edge. When the robot approaches the book with a stiff position controller, unsurprisingly, the book topples. However, if the robot turns off the motors (zero commanded stiffness) the fingers can comply to the book maintaining its vertical state. The transparency also allows for variable impedance control. Varying the PD gains allows us to control the stiffness and damping of the fingers.

4.3 Infrastructure and Software

The DDHand is mounted on an ABB IRB120 robot with a ATI mini40 F/T sensor mounted inline between the hand and the robot wrist. Low-latency communication is essential in the implementation of dynamic behaviors and feedback strategies mentioned in Chapter 5.

For a high-performance control implementation, we employ ROS Control [100] to keep all control computations running in one thread. The architecture for software and hardware infrastructure is shown in Fig. 4.5.

The hardware interface layer for the hand, arm, and F/T sensor communicate with the hardware through their specified protocols. The hand uses an asynchronous SocketCAN interface to communicate joint level position setpoint and gains with the two fingers at 10kHz. The arm uses ABB's proprietary Externally Guided Motion (EGM) protocol over UDP at 250Hz with an average 25ms latency. The force/torque sensor interface communicated with a NetF/T data acquisition module over UDP at 3kHz

The DDHand Controllers are modular, i.e., they can be switched, started or stopped while the process is running and allow seamless switching. Each controllers is implemented as a shared object and can borrow a subset of interfaces. Two controllers cannot write to the hardware interfaces simultaneously.

A simulation of the hardware is implemented in ROS/Gazebo with DART as a backend. The simulation presents the same interface to the controllers to make the transition from testing in simulation to hardware effortless.

The hand-arm system has a combined ten degrees of freedom. In most cases, our task space is limited to the two fingertips. We assume the fingertips to be point contacts. A total of six freedoms are specified by the two point contacts: an (x, y, z) tuple for each fingertip. That leaves four redundant degrees of freedom leading to an infinite number of solutions for the inverse kinematics (IK). We express these four freedoms as geometric parameters x, y, θ, ϕ to locate the palm of the hand with respect to the fingertips. ϕ is the angle of the gripper plane normal with respect to the vertical; x, y, θ define the transform between the center of the palm and the fingertip frame within the gripper plane. The fingertip frame is located at the midpoint of the line joining the fingertips with the x-axis along the line joining the fingertips.

4.4 Discussion

In this chapter, we have discussed the design and control of direct-drive hands. We have looked at two revisions of the design the DDHand [11] and the Dexterous DDHand.

Even though motor technology has come a long way, direct-drive is still accompanied by the possible loss of torque due to lower torque densities and lack of torque multiplying transmissions. We envision three ways to work around these lower torques:

Thermal Management When humans try to pick up something heavier than what we are used to, we either sweat to manage the heat and if that is not enough, we put the

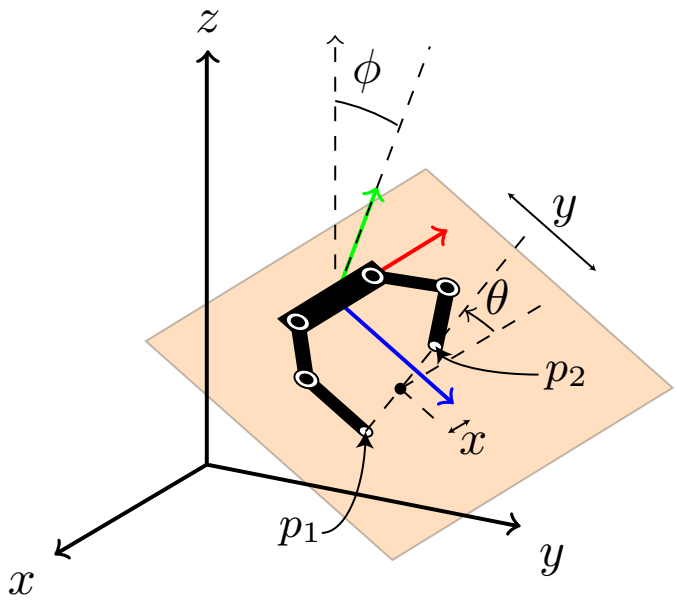


Figure 4.6: Redundant coordinates for the 10 degree of freedom robot-arm system. The Red, Green and Blue axes are define the x, y and z axis of the end effector frame of the robot. The fingertips p_1 and p_2 are specified by the task and define 6 degrees of freedom of the system. The Gripper plane (orange) is tiled with respect to the global coordinate frame by an angle ϕ. The location (x, y, θ) of the midpoint of the line joining p_1 and p_2 define the final 3 coordinates of the system.

thing down briefly. Motors, too, can be driven over their nominal operating limits as long as the temperatures are managed intelligently. This can be done either through active or passive cooling (for example, blowing air over the motors), as in [7, 126], or by intelligent controllers that sense the motor state and take breaks to prevent permanent motor damage, as in [31, 105, 125].

Kinematic Singularities Another strategy humans employ when carrying heavy things is the use of singularities. For example, while carrying a bag of groceries we tend to lock out our arm straight down to let the structure take most of the weight.

The kinematics of the DDHand finger linkages are nonlinear, and so characteristics such as compliance and force limits vary with operating point (i.e. with the choice of hand pose and grasp configuration). By choosing a hand pose it is possible to put the fingers at singularities or at travel limits to provide high forces, or the opposite to provide high compliance [17].

Directional Rigidity Conventional manipulators try to be stiff in all directions. Isotropic behavior is an explicit design objective of some grippers, for example Salisbury's milestone design [111]. The conventional approach is essentially a know-nothing approach, reducing the required planning and control intelligence, but also reducing the variety of mechanical intelligence available. Further, universal isotropic stiffness leads to stiff and heavy hands. The DDHand adopts the opposite approach: deliberately anisotropic stiffness (Four DOF arms like the Delta or SCARA arm apply a similar design trick.) It is rigid in some directions and compliant in others. Specifically, the two degree of freedom fingers are constrained to be planar and can maintain rigidity in the out-of-plane directions (by pushing against the structure and bearings). By intelligently planning grasp and transfer motions, we can achieve lower peak gripping forces.

With these strategies the reduced torque of direct-drive can be mitigated, while still maintaining the transparency, force bandwidth, and speed. Grippers like the DDHand can open up new capabilities and behaviors that allow a robot to naturally react to the environment instead of imposing its will on it.

Chapter 5

Move-until-touch Behaviors

Introduction

A direct-drive gripper like the DDHand enables high-performance reactive behaviors that are not possible with other actuation schemes.

Move-until-touch is a common motion primitive applied when making contact for robotic systems with touch sensing. This primitive behavior has shown to be useful in centering grasps [49, 95, 96].

In this chapter we explore the smack-and-snatch behavior which takes advantage of the improvements in speed, transparency and bandwidth as a result of direct-drive actuation. We then compare robotic systems on the collision reflex metric, first comparing direct-drive and geared 1 degrees of freedom actuators and then the DDHand to a COTS Schunk gripper.

5.1 Smack-and-snatch Grasping with the DDhand

We apply the move-until-touch behavior to the DDHand and show performance improvements that arise from its high-transparency degrees of freedom. We apply the move-until-touch primitive in a direction orthogonal to the grasp axis: the smack-and-snatch grasp Fig. 5.1).

The goal is to quickly acquire an object at rest on a table. The table's height is not known precisely. The gripper smacks the table and snatches the object so quickly that the arm need not slow down at all. (A grasp called a "snatch" was demonstrated in [78], using inertial forces to acquire an object without squeezing it). A traditional industrial robot and gripper would have to approach the table very slowly, detecting contact with a force/torque sensor, contact sensor, or by monitoring servo error to avoid crushing the finger into the surface.

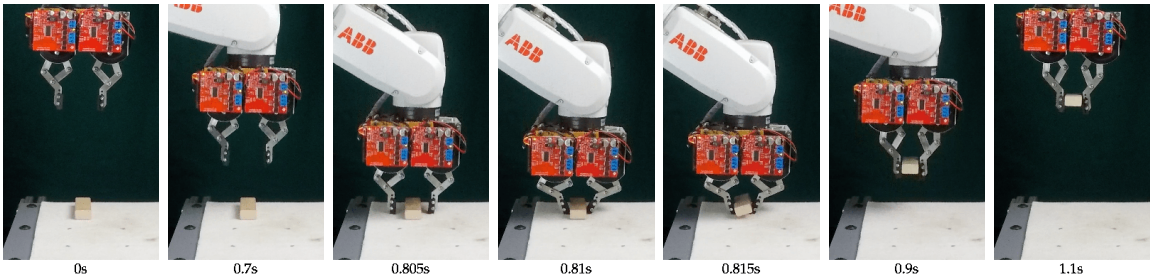


Figure 5.1: The DDHand executing the "smack and snatch" behavior. The manipulator is uncertain of the object pose and the table height. The arm accelerates the hand towards the object; fingers detect contact with the table then slide along the surface to locate the object; the arm simultaneously decelerates; the fingers grab the object while the arm accelerates upwards. The grasping maneuver is completed within one second, starting and ending at rest.

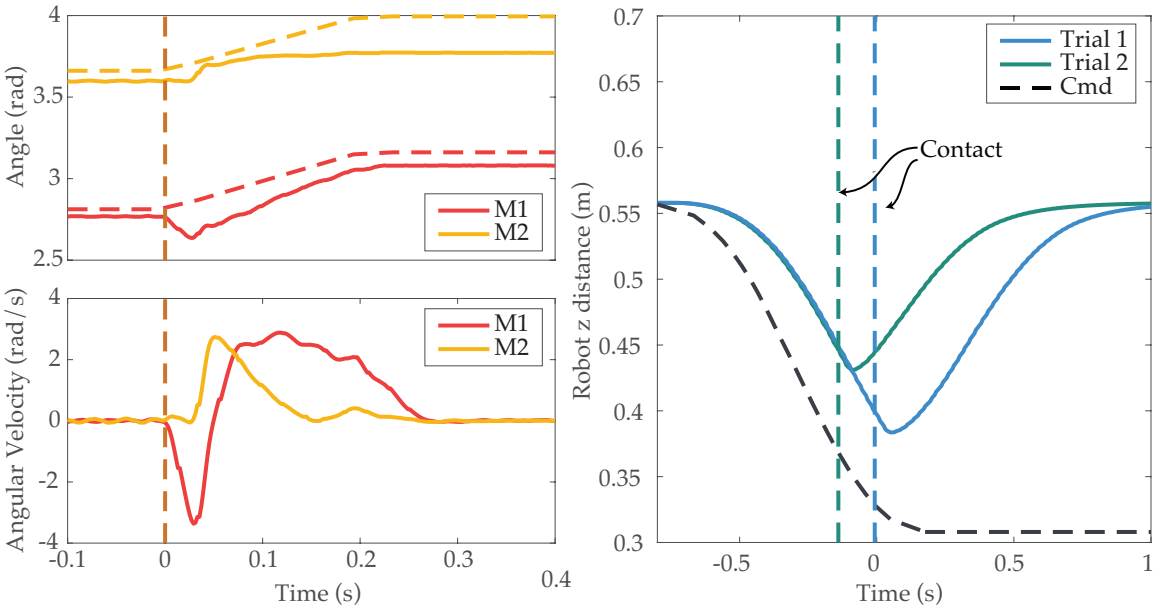


Figure 5.2: Left: Evolution of the joint angles of one finger as it makes contact with the surface and begins the snatch behavior. Right: The evolution of the Z coordinate of the robot tool frame. The two trials show the behavior with two different table heights. For all plots, the dotted lines are the commanded reference trajectory. For the robot (right), the dotted line is the unassuming command which would execute if there was no contact detected with the table.

To begin this task, the robot arm approaches a small object on a table at high speed. The position of the object and height of the table are not known exactly. The fingers make contact with the table first, but their low mass and low reflected inertia ensures that the impulse is low and the hand does not get damaged. When the hand detects the contact the arm's motion can be altered by an upward acceleration that slows and eventually reverses the approach. While the arm is still in motion, the fingertips can track the table surface as

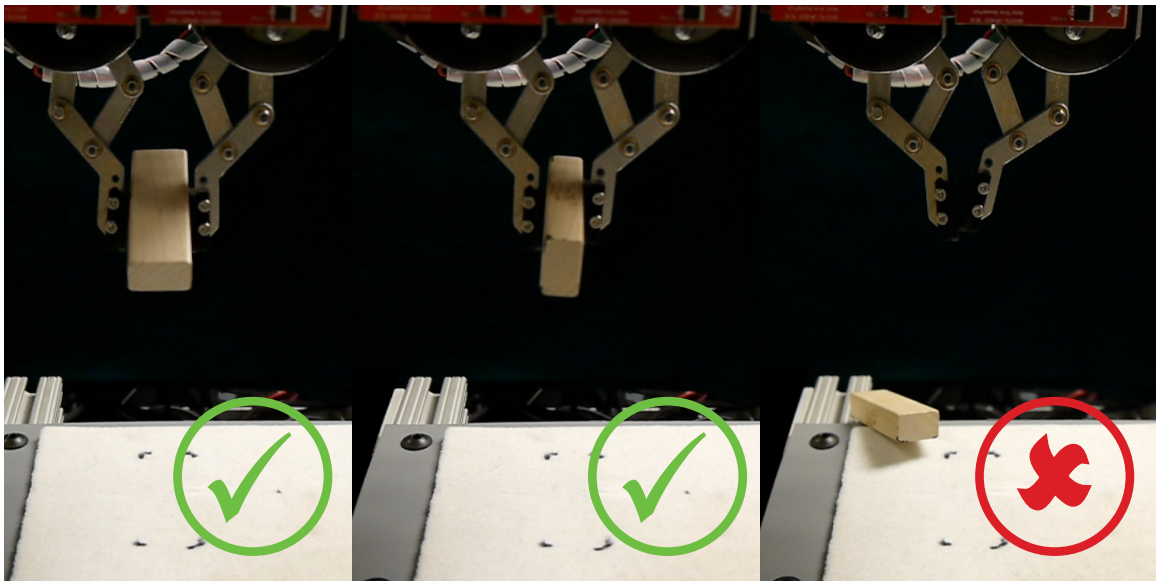


Figure 5.3: The figure shows the final state of the block in three out of fifty trials. The first two figures from the left show successful grasps and the rightmost figure shows a failure.

they close, and continue to monitor force to detect contact with the object. The behavior is implemented with an ABB IRB120 robot and the DDHand. The dynamic nature of this behavior requires the control and execution of trajectories on the hand and the arm to be tightly coupled. To achieve the tight timing requirements, the arm is controlled via its Externally Guided Motion (EGM) interface with a communication rate of 250Hz with a 25ms latency. The latency combined with the finite deceleration of the arm require the fingers to be able to absorb the impact forces. During this deceleration, the arm travels approximately 13mm. This travel is absorbed by the fingers. Post-impact, a manually coded trajectory is executed on the hand with maximum torque limits to execute the snatch phase of the grasp.

A stop motion image sequence is shown in Fig. 5.1. The behavior is designed to be agnostic to the height of the object. The results from executions at different table heights is shown in Fig. 5.2.

The smack and snatch behavior was repeated fifty times to get a preliminary estimate of robustness. All fifty trials is included in the video attachment accompanying [11]. A trial was labeled successful when the block transitioned from its initial state to a stable pinch grasp in the hand. The final pose of the block is ignored and can lie in one of two configurations as shown in Fig. 5.3. Out of fifty attempts, the behavior was successful forty-three times and failed to grasp the block seven times (success rate of 86%.) The failures may be attributed to two factors:

1. On compression of the fingers in the approach phase, a part of the energy is transferred to the table as spring potential energy. This energy, when released, causes an upward velocity of the block making the grasp unstable.

2. If the object is not centered within the fingers, the momentum of the finger that touches the object first is partially transferred to the block. This causes the block to bounce around between the fingers.

These failure modes are equally likely. Out of seven, four failures are due to the first mode and three failures to the second.

5.2 Collision Reflex Experiments

The collision reflex metric described in Chapter 3 has similarities with the move-until-touch behavior. The two phases of initial impact, sensing and reaction closely resemble the move-until-touch behavior where the reaction is not just to stop but to accelerate until the normal force at the constraint is zero.

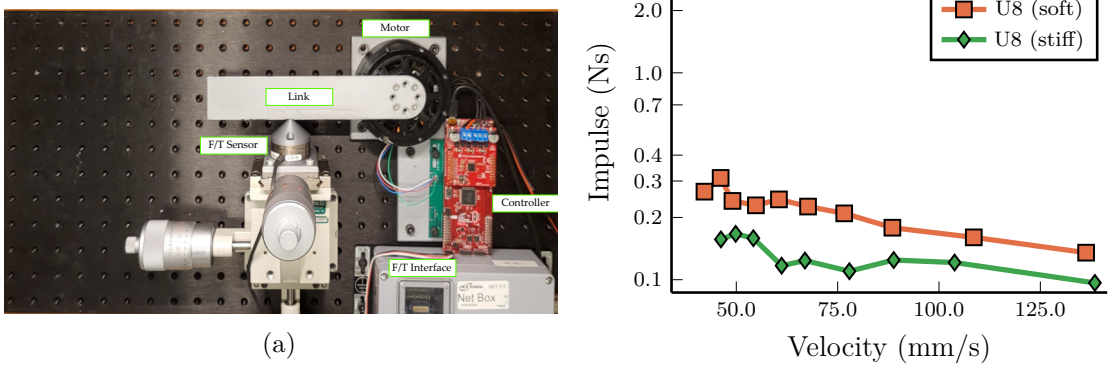
Through experiments we attempt to ground the collision reflex metric in reality and discover deviations from it. We implement the collision reflex experiment: systems capable of collision detection undergo an unexpected collision with a force-torque sensor (ATI mini40). The force signal from the sensor is captured and the collision reflex is calculated as the area under the force-time curve. Fig. 3.5 (bottom) shows a typical trace of a force-time curve from an experiment. The collision reflex is measured for: (Section 5.2.1) 1D collisions – comparing a single joint geared and direct-drive actuator; and (Section 5.2.2) Gripper collisions – comparing the Schunk WSG-50 gripper and the DDHand.

5.2.1 Collision Reflex in One-dimension

We test the collision reflex of a direct-drive and geared actuator in a one-dimensional collision. The direct-drive actuator is a T-motor U8 (a BLDC motor used, e.g., on the Ghost Robotics Minitaur[63]). The geared actuator is the T-motor A8 which is a U8 motor augmented with a 9:1 planetary gearbox. Both motors are controlled by a TI instaspin motor controller running custom firmware with a RLS RM-08 magnetic encoder for position sensing and commutation. Fig. 5.4a shows the setup for this experiment. The collision is set up to occur at link angle of 0 rad with a characteristic length of 114.3 mm.

The behavior for this experiment implements the sensing phase as a move-until-touch action. The link starts at initial pose $q_i = -\pi/2$ and executes a spline trajectory to final pose $q_f = \pi/2$ for time t. The velocity of impact is varied in proportion to the trajectory time. A threshold on force at the link, as measured by motor current, provides collision detection capabilities. Once collision is sensed, the motor is commanded to q_i as fast as the system allows. For the short compression distance of the force sensor, the motor can be assumed to operate at constant acceleration.

The evolution of force over the impact is captured at 3500 Hz over a UDP communication link. This evolution is shown in Fig. 3.5 (bottom) for a U8 motor and more examples are shown in Fig. 5.5 for the U8 in low and high gain stiffness, as well as the A8. Changing



(a) (b)

Figure 5.4: (a) Experimental Setup for 1D experiments. The figure shows a gear motor (T-motor A8 with a transmission ratio of 9:1) connected to a link which impacts an ATI mini40 force-torque sensor. The motor is driven by a TI instaspin enabled C2000 microcontroller with a BOOSTXL-DRV8305 3 phase driver. The experiment is repeated with a T-motor U8 in a direct-drive configuration. (b) The collision reflex impulse with varying velocity for the A8 (geared) and U8 (direct-drive). The U8 experiments are duplicated with two levels of software stiffness. The y-axis is shown as a log plot to better show the separation between the actuator configurations.

stiffness for the A8 has little effect on the output due to higher reflected inertias. We estimate the parameters of the collision reflex model (m_f, k_s, F_s and a) for each collision using a one-shot least squares fit of the force-time profile. Mechanical Stiffness (k_m) is assumed to be 2×10^7 N/m which is the mechanical stiffness of the F/T sensor along the sensing direction. The normal distributions of these parameters are reported in Table 5.1.

The finger mass m_f for the U8 in high and low stiffness configurations is ≈ 0.12 kg while that of the A8 is higher due to added inertia of the gearbox and the reflected inertia from the transmission. The stiffness increases as the U8 switches from low and high stiffness modes. The higher stiffness of the A8 is due to reduced backdrivability form the transmission. As the threshold F_s is applied on the motor side of the transmission for proprioceptive sensing,

		U8		A8
Parameter	Unit	Low k (Soft)	High k (Stiff)	
Finger mass (m_f)	kg	0.10(2)	0.145(70)	20.04(240)
Stiffness (k_s)	N mm^{-1}	0.440(70)	1.07(41)	23.7(63)
Threshold (F_s)	N	3.00(34)	3.03(33)	40.85(257)
Acceleration (a)	mm s^{-2}	(4.7(31))E-2	7.0(150)	3.1(12)

Table 5.1: Estimated parameters from the 1-shot model fit for the Direct-drive (U8) and geared (A8) actuators.

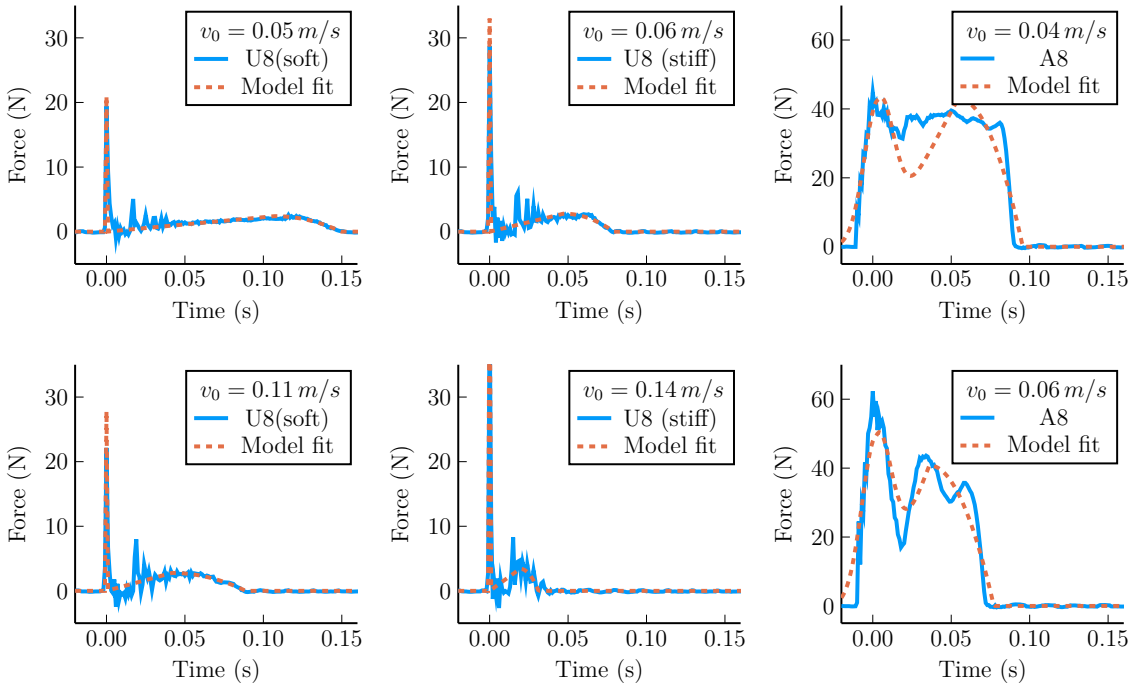


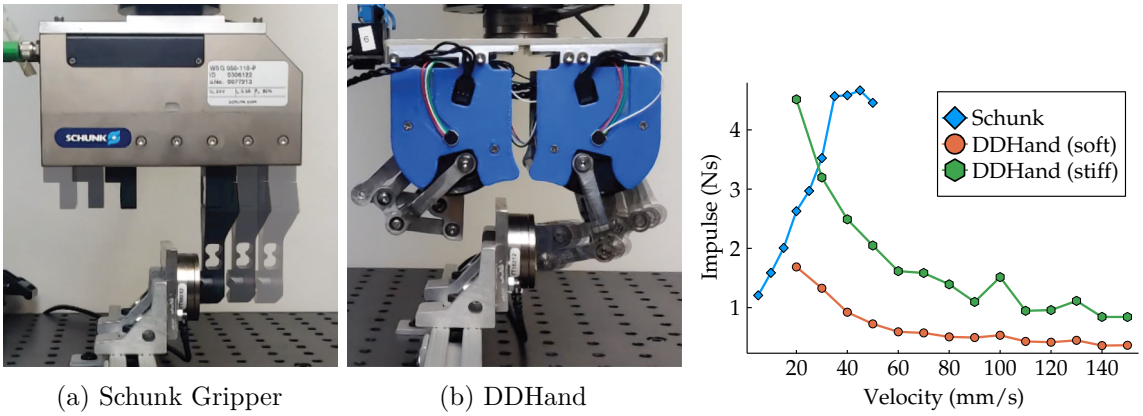
Figure 5.5: Force measurements over the duration of the impulse for: (left) U8 (direct-drive, soft), (center) U8 (direct-drive, stiff) and the A8 (geared) actuator configurations. The dotted line in each shows the collision reflex model fit to the data. Table 5.1 shows the uncertainty in the parameter fit for each system.

the U8 actuator detects collision at $\approx 3\,\text{N}$ force while the A8 detects collision at $\approx 40\,\text{N}$. Fig. 5.5 shows the model fit to measure these parameters. Note that the model fit for the A8 motor is not as accurate due to deviations in the model from reality caused by additional inertia and mechanical delays due to the added transmission.

The total impulse imparted in the collision is determined by numerical integration of force over time using adaptive Gauss-Kronrod quadrature [55] as shown in Fig. 5.4b. We see that for both direct-drive (U8, $N = 1$) and quasi-direct-drive (A8, $N = 9$) actuator configurations the impulse decreases as we increase velocity in this regime. It is not surprising that the impulse for the A8 is much higher than that of the U8 due to higher reflected inertias of the U8 rotor through the transmission. Both systems are operating at $v_0 < v^*$ (Eq. (3.22)) in the above experiments.

5.2.2 Comparing the Collision Reflex of Robot Hands

We measure the collision reflex of two robot hands: The Schunk WSG-50 gripper and the Dexterous DDHand [11]. The WSG-50 end-effector is a popular COTS offering from Schunk. It has a parallel jaw configuration driven by a electric servo motor with a ball screw transmission. It has a stroke of up to 110 mm and a peak grip force of 80 N. The Dexterous DDHand is a 4 degree-of-freedom direct-drive end-effector. It is actuated by a



(a) Schunk Gripper (b) DDHand

Figure 5.6: The collision reflex experiment for the DDHand [11] and Schunk end-effectors. (Top): The fingertips are impacted against a force sensor with a move-until-touch and react control loop. (Bottom): The total impulse for varying pre-impact velocities.

set of T-motor GB54-2 motors connected to 3 parallelogram linkages that mimic a two-link serial robot while keeping the fingertips parallel.

Fig. 5.6 shows the setup for this experiment. The two grippers were mounted on a stationary ABB IRB 120 robot above a force-torque sensor (ATI Mini40.) A move-until touch controller was implemented on both robots with proprioceptive collision sensing. As the control architecture for the schunk gripper is abstracted away from the user, a workaround using the gripper's close until force command was used with one of the fingertips removed. Once the action reported successful completion, the robot was instructed to open at it's maximum speed.

For the DDHand, the collision sensing was implemented as an error threshold on the position controller, i.e. if the position tracking for any of the joints was lagging behind the command, the finger was declared in collision. Again, once the collision was detected, the robot was instructed to open at its maximum speed.

Data was collected at pre-impact velocities up to 55 mm/s for the Schunk gripper and up to 150 mm/s for the DDHand. The force-time data was integrated using adaptive Gauss-Kronrod quadrature and the results are presented in Fig. 5.6.

The DDHand, being direct-drive, is more transparent. It follows the unintuitive trend of lower impulse at higher pre-impact velocities (as all attempted collisions for the DDHand exist in the space of collisions where $v_0 < v^*$). The Schunk gripper on the other hand shows a more conventional linear trend due to the impulse from the initial impact phase dominating. All attempted collisions for the Schunk gripper exist in the sa the space of collisions where $v_0 > v^*$.

5.3 Discussion

5.3.1 Smack-and-snatch Grasping

The improved bandwidth, speed, and transparency with direct-drive actuation are well demonstrated by the smack and snatch task which is hard to realize with conventional grippers. Consider, for example, the Schunk gripper [113] and the Robotiq gripper [104] as examples of geared grippers with strain-gauges and series-elastic actuation, respectively. The Schunk gripper will fail to execute this behavior at the approach phase as it lacks the ability to manage the impulse imparted during collision with the table leading to damage to the fingers or strain-gauge depending on their relative strength. For the Robotiq gripper, the lower stiffness of the passive compliance can manage imparted impulse during the approach phase but the high gear ratio will prevent the fingers to close around the object fast enough to snatch it.

There is still scope to improve the robustness of the smack and snatch grasping behavior with the DDHand. The first of the two failure modes discussed in Section 5.1 can be resolved by ensuring the table surface is not compliant and prevent the energy transfer to the block or reducing the stiffness of the fingers post-impact. The second failure mode is due to uncertainty of the 2D pose of the block on the table. Better state estimation can lessen the presence of this mode. Integrating intrinsic sensing is also desirable. The fingers can monitor the motor torques during the snatch phase and decelerate the finger that makes contact with the block first.

5.3.2 Collision Reflex Metric

The simple mass-spring-mass collision model used to analyze collision reflex has some limitations. For example, it doesn't account for the nonlinear behavior of transmissions that stem from backlash, friction, and gear inertias. The model also fails to capture time discretization and delays and in communication, although those could be added in. In both our real-world experiments, the behavior of the geared systems deviated from the ideal response. It is unclear how much of this deviation can be accounted for with a first principles approach. These shortcomings make it more difficult to use the model for learning parameters and accurate prediction of the collision reflex from a few data points. The model parameters learned from experiments (Table 5.1) have non-trivial uncertainties associated with them. Future work should address the limitations of the simple model in order to enable more accurate predictions.

The inaccuracies in modeling the real world do not take away from the usefulness of the collision reflex metric. The analysis in this chapter has shown some unintuitive results for closed-loop robotic systems. Most notably, there is an optimal velocity for minimizing impulses during collision. The metric also captures the trends that we see in real-world

impulses during collision. The use of this metric is relevant in the design and motion planning of robots where minimization of impulses is essential in a collision. We envision the application of this work in Robotic manipulation and locomotion on land and even in space.

Chapter 6

Contact Localization using Velocity Constraints

Introduction

In unstructured environments robots must deal extensively with the making and breaking of contact with uncertain or unknown object shapes and poses. This makes contact localization a vital skill. For example, Fig. 6.1 shows a legged robot walking up stairs, with one of its legs in contact with the edge of a step. In this scenario, noisy estimation of the height of a stair can lead to unexpected contact between the edge and the leg causing the robot to trip. Another example is an industrial robot gripper with planar fingers approaching an object to grasp or estimate its surface, Fig. 6.9.

In this chapter, we propose a generalized extension to the method used by Barasuol et al. [9] for planar velocity-based contact localization, summarized in Fig. 6.1. The method is based on the observation that if a point is in contact with a rigid body, its velocity in the direction of the surface normal must be zero, i.e., its velocity is perpendicular to the surface normal. Calculating the instantaneous velocity at each point on the surface will yield a set of candidate points.

The main requirement for this method to work is that the robot be rigid and have accurate position and velocity measurements on the collision link. Any motion due to collision is directly transmitted to the link's position/velocity sensors. The robots used for this work are transparent in the conversion of external force to motion due to their direct-drive joints. That is, any external forces are efficiently relayed to motion at the joints that can be picked up by joint encoders. Adding mechanical or software compliance to the articulated joints are alternate ways to achieve high transparency. Adding inertial measurement units can also provide the information needed to compute collisions with the velocity contact localization method.

We analyze this method and claim that it has the following properties: 1) the method

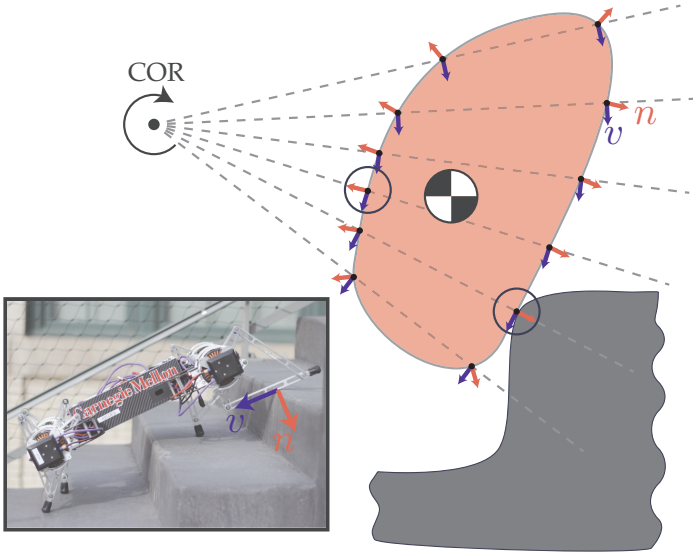


Figure 6.1: A velocity-based contact localization scheme. Contact is localized to a set of candidate points (circled) where the surface normal n is co-linear with the line joining the candidate point to the center of rotation (COR). Equivalently, the velocity at the candidate point v must be perpendicular to the surface normal. Inset: Example application of a quadruped robot on a flight of stairs.

provides an instantaneous estimate of contact point locations. 2) The method does not require an accurate dynamical model of the robot. 3) The method can be implemented on existing robots without any additional sensors beyond joint position or velocity measurements. 4) The method uses a velocity constraint to produce a set of possible contact points. For general planar systems, including the legged robot and gripper of interest to us, this is sufficient to isolate the contact point to a 0-dimensional set (i.e. an individual point or, if there is ambiguity in shape, a set of possible points). For spatial systems, this produces a 1-dimensional set of possible contact points (one or multiple curves). In general, it produces a codimension 1 set of points, i.e., an n-dimensional surface in collision should produce an n-1 dimensional set of possibilities. This can be reduced down to a single contact point through filtering, assuming frictional contact, or adding additional dynamic constraints.

We evaluate the performance of the velocity-based method and compare it with position and torque-based methods in a simulation of a five-link planar robot. We also demonstrate the velocity based method in hardware experiments with a legged robot (Minitaur [63]) and a two-finger gripper (DD Hand as described in Chapter 4 and [11]). We then show a proof of concept to extend the planar method to 3D with the help of a particle filter.

6.1 Related Work

It is common to localize unknown contacts with exteroceptive sensors such as LIDAR, structured light, and stereo cameras. However, these sensors often do not give the accuracy

needed to perform precise tasks such as fine manipulation or maneuvering in a tight space. Furthermore, in cases with poor lighting, feature starvation, or occlusion, these sensors may fail to detect obstacles, leaving the robot unable to anticipate impending collisions. This drives the need for fast and robust contact sensing and localization.

One popular method to achieve this employs tactile sensors, e.g. [25, 119]. A sensitive skin is applied to the surface of the robot that can then measure contact forces and/or locations. Unfortunately, the implementation of tactile sensors leads to undesired design constraints, higher cost, and limited material choices. In fact, most robots utilize sensorized skin only on areas with high probability of contact, such as fingertips and feet, leaving the rest of their body uncovered (with some notable exceptions, e.g. [52, 92, 134]). Tactile sensors are also limited in their sensitivity and spatial resolution.

Methods based on proprioceptive sensing can overcome some limitations of local tactile sensing. Proprioceptive sensors measure the internal state of the robot (joint positions, velocities, and torques) and can ideally detect contact anywhere on the robot structure. Fortunately, most robots already measure their internal state for control using joint encoders and torque sensors so no additional hardware is required. Position, velocity, and torque signals can all be used to localize contacts. Torque-based contact localization methods are the most popular [81, 127]. However, a major drawback is that they require accurate dynamics to estimate external forces. Uncertainty in parameters such as weight distribution, friction, and damping limit the accuracy of these methods.

For a suitable subset of robot problems, just position [57, 69] or velocity [9] can be sufficient to localize contact. Position-based methods require position measurements to be spaced out in time and assume stationary point contacts. This makes them unfavorable for moving contacts such as when the contact is rolling or sliding. Velocity-based contact localization methods are less common, but recent work on the HyQ robot [9] suggests they hold merit.

6.2 Problem Setup & Notation

The chapter follows modified notations from [90]. All vectors are denoted with lowercase boldface fonts (e.g. $\boldsymbol{a}$ or $\boldsymbol{c}$), all matrices are uppercase boldface font (e.g. $\boldsymbol{J}$ or $\boldsymbol{Ad}$), all sets are written in uppercase font (e.g. B or C), and all frames are written in roman type style (e.g. P or L.) The terms defined in this section are shown in Fig. 6.2.

We consider a robot with m actuators and n rigid links. We assume that the robot motion is constrained to be planar (although we relax this assumption in [129]). We assume that the robot has an accurate knowledge of its geometry, and can measure its actuator positions and velocities, which are denoted as $\boldsymbol{q} = (q_1, \ldots, q_m)^T$ and $\dot{\boldsymbol{q}} = (\dot{q_1}, \ldots, \dot{q_m})^T$.

We also assume that the robot has an accurate estimate of its body velocity twist $\boldsymbol{v}^b_{\mathrm{S,P}}$, where S is a frame fixed to the world, and P is a frame fixed to the robot's body. We define

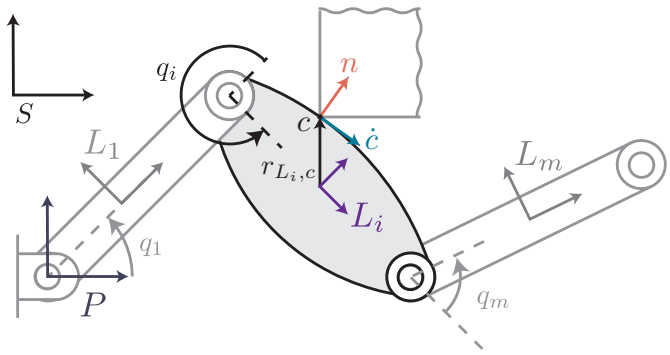


Figure 6.2: Schematic of a generic robot with m actuators showing the notation used.

n frames $\mathrm{L}_1, \mathrm{L}_2, ..., \mathrm{L}_n$ that are fixed to each link of the robot (such that the transformation between P and L_i is described entirely by $\boldsymbol{q}$). At time t_0 the i^{th} link of the robot collides with a stationary foreign object and is sliding along the object until time $t_f > t_0$. We assume that the robot is compliant enough that it does not come to a complete rest after colliding with the object, but continues to roll or slide along the object ($\|\dot{\boldsymbol{q}}(t)\| + \|\boldsymbol{v}^b_{\mathrm{S,P}}\| \neq 0 \ \forall t \in [t_0, t_f]$). We define B as the set of all points on the surface of link i at which contact could have occurred. Note that these could be on any surface of the link. It is also possible to pre-filter this set for surface points that are not expected to be in contact. For a point $\boldsymbol{c} \in B$, the vector $\boldsymbol{r}_{\mathrm{L}_i,\boldsymbol{c}}$ is defined as the coordinates of point $\boldsymbol{c}$ in the L_i frame. The vector $\boldsymbol{n}_{\mathrm{L}_i,\boldsymbol{c}}$ is defined as the unit vector normal to link i's surface and pointed outwards at point $\boldsymbol{c}$, also in the L_i frame.

We define $\boldsymbol{J}_i$ as the body Jacobian that maps actuator velocities, $\dot{\boldsymbol{q}}$, to the body velocity twist $\boldsymbol{v}^b_{\mathrm{P,L}_i}$ of frame L_i relative to frame P,

$$\boldsymbol{v}^b_{P,L_i} = \boldsymbol{J}_i \dot{\boldsymbol{q}}. \tag{6.1}$$

$\boldsymbol{J}_i$ only explains the internal motion of the frame L_i in the robot's base frame. To account for it's spatial motion, we also need to consider $\boldsymbol{v}^b_{\mathrm{S,P}}$.

Given the homogeneous transformation matrix $\boldsymbol{H}_{1,2}$ between two frames 1 and 2, with rotation matrix $\boldsymbol{R}_{1,2}$ and translation vector $\boldsymbol{r}_{1,2}$, define the adjoint matrix $\boldsymbol{Ad}_{\boldsymbol{H}_{1,2}}$ as,

$$\boldsymbol{Ad}_{\boldsymbol{H}_{1,2}} = \begin{bmatrix} \boldsymbol{R}_{1,2} & \hat{\boldsymbol{r}}_{1,2}\boldsymbol{R}_{1,2} \\ 0 & \boldsymbol{R}_{1,2} \end{bmatrix}, \tag{6.2}$$

where $\hat{\boldsymbol{r}}_{1,2}$ is the skew symmetric matrix of vector $\boldsymbol{r}_{1,2}$. For more information on the adjoint operator in screw theory, the reader is referred to [90].

Upon contact with the foreign object, the robot must localize the point of contact, that is, compute $\boldsymbol{r}_{\mathrm{L}_i,\boldsymbol{c}}$.

6.3 Review of proprioceptive methods

We compare our method of contact localization against two other proprioceptive methods. A brief review of these methods follows.

6.3.1 Position-based Contact Localization

Position-based methods depend on self-posture changing motions [57]. The robot is assumed to have collided with a stationary wedged-shaped object. Hence, the point of contact remains stationary in the world frame S. To estimate the location of contact, joint positions at two instances $[t-\Delta, t]$ during the contact time window $[t_0, t_f]$ are recorded. The method is generally accompanied by an exploratory motion to space out the two measurements. The intersection of the surface of link i at these positions localizes the contact.

If the object breaks the assumption that it is a wedge (a curved object, for example) the estimated contact location will have some error and may be off of the object as the contact location will have moved between the two measurements. To improve the accuracy for such cases, the measurements can be moved closer together in time which will approach the proposed velocity-based method. Note that as the measurements become closer, the estimated contact location will become increasingly noisy.

6.3.2 Torque-based Contact Localization

This method, also known as intrinsic tactile sensing [14], assumes that an unknown linear force, $\boldsymbol{f}_{ext}$, and zero moment is applied at the contact point due to collision with the foreign object. This results in the wrench, $[\boldsymbol{f}_{ext}^T, \boldsymbol{0}^T]^T$, in frame L_i, applied at the point $\boldsymbol{c}$. Applying this wrench at the point $\boldsymbol{c}$ is equivalent to applying the wrench $\boldsymbol{f}_i$ at frame L_i, where $\boldsymbol{f}_i$ is given by,

$$\boldsymbol{f}_i = \begin{bmatrix} \boldsymbol{f}_{ext} \\ \boldsymbol{r}_{\mathrm{L}_i,\boldsymbol{c}} \times \boldsymbol{f}_{ext} \end{bmatrix}. \tag{6.3}$$

It is also equivalent to an external torque, $\boldsymbol{\tau}_{ext} \in \mathbb{R}^m$ applied to the robot's actuators,

$$\boldsymbol{\tau}_{ext} = \boldsymbol{J}_i^T \boldsymbol{f}_i. \tag{6.4}$$

With a momentum observer, velocity observer, or another collision monitoring method (as in [39]), the external torque, $\boldsymbol{\tau}_{ext}$ can be estimated.

For this method, it is required that the robot can measure these actuator torques for which an accurate dynamical model of the robot might be required. With the estimate of the external torques $\boldsymbol{\tau}_{ext}$, Eq. (6.3) and (6.4) can be solved for $\boldsymbol{r}_{\mathrm{L}_i,\boldsymbol{c}}$, and $\boldsymbol{f}_{ext}$.

For a robot in the plane, the surface of link i is one dimensional. Since the contact point lies on this one dimensional surface, $\boldsymbol{r}_{\mathrm{L}_i,\boldsymbol{c}}$ can be reduced to one unknown variable. As $\boldsymbol{f}_{ext}$ has unknown magnitude and direction, $\boldsymbol{\tau}_{ext}$ needs to be at least 3 elements long

for the problem to have a unique solution. This implies the robot must have at least three degrees of freedom (DOF). If a further assumption of frictionless contact can be made, $\boldsymbol{f}_{ext}$ can be reduced to one unknown variable and this method can generate a unique solution on a 2 DOF robot. Similarly a spatial robot must have at least 5 DOF, or 3 DOF in the frictionless case, to generate a unique solution.

6.4 Proposed Velocity-based Method

The torque-based method uses the transpose of the Jacobian to relate end-effector force to joint torques. Here, we propose a method that utilizes its dual relationship: use the Jacobian to map joint velocities to end-effector velocity. That, combined with the velocity constraint enforced by a collision, can localize the contact point.

We begin by summarizing the assumptions that need to hold to use velocity for accurate contact localization.

1. The robot must be compliant and have at least one degree of freedom after applying the constraint. That implies that the robot does not come to a complete stop after collision.
2. Contact detection is solved, i.e., we know the time of collision and which link has collided with the external constraint.
3. A good estimate of body position and velocity of the collision link is available. The better this estimate, the more accurate the contact localization.

To derive the location of the contact point, we need to find the linear velocity of a point $\boldsymbol{c}$ on the surface of the link L_i. Let us first express the body velocity twist of frame L_i in terms of the generalized coordinates,

$$\boldsymbol{v}^b_{S,L_i} = \boldsymbol{Ad}_{\boldsymbol{H}^{-1}_{\mathrm{P},\mathrm{L}_i}} \boldsymbol{v}^b_{\mathrm{S,P}} + \boldsymbol{J}_i \dot{\boldsymbol{q}}. \tag{6.5}$$

With this, we can express the linear velocity of point $\boldsymbol{c}$, in terms of the velocity of frame L_i at time t as,

$$\dot{\boldsymbol{c}}(t) = \begin{bmatrix} I & -\hat{\boldsymbol{r}}_{\mathrm{L}_i,\boldsymbol{c}} \end{bmatrix} \boldsymbol{v}^b_{\mathrm{S},\mathrm{L}_i}(t). \tag{6.6}$$

Based on this, the scalar velocity in the normal direction, $\dot{c}_n(t)$, of a point $\boldsymbol{c}$ at time t is,

$$\dot{c}_n(t) = \boldsymbol{n}_{\mathrm{L}_i,\boldsymbol{c}} \cdot \dot{\boldsymbol{c}}(t), \tag{6.7}$$

where $\boldsymbol{n}_{\mathrm{L}_i,\boldsymbol{c}}$ is the surface normal of link L_i at $\boldsymbol{c}$.

6.4.1 The Method

There are two velocity constraints that must hold at the true contact point ($\boldsymbol{c}^*$). First, during contact, $\boldsymbol{c}^*$ must have zero velocity in the direction normal to the link surface,

$$\dot{c_n^*}(t) = 0 \quad \forall t \in [t_0, t_f]. \tag{6.8}$$

This is necessary for persistent contact between the link and the object. For the planar case, this constraint is equivalent to having the line from the center of rotation to the point $\boldsymbol{c}^*$ be perpendicular to the robot's surface, as shown in Fig. 6.1.

Second, at the instant before contact, denoted as t_0^-, the point of initial contact must have a positive velocity along the surface normal of the link,

$$\dot{c_n^*}(t_0^-) > 0. \tag{6.9}$$

Thus, to localize a contact, the proposed method simply identifies the set of possible contact points, denoted C, which contains candidate points that satisfy both constraints. An example of computing the set C for a case with simple link geometry is given below. An algorithm to compute C when there is complex link geometry is detailed in [129].

The codimension of C in B is one. For the planar case, since the set B is one dimensional, the set C is zero dimensional. In many planar cases, C will contain a unique possible contact point at time t_0. For a spatial robot, the set B is two dimensional, so the set C is one dimensional. Reducing contact location ambiguity in cases where C contains more than one point is discussed in [129].

6.4.2 Simple Geometry Example

We now apply this method to an example robot with simple link geometry. Consider a Minitaur robot that has collided with the edge of a stair, Fig. 6.1. Using one of the methods described in [39], the robot has detected that the collision occurred at time t_0 and lasts until time t_f. Fig. 6.3 shows the link of the robot that has made contact. The portion of this link's surface that is expected to make contact, B, is highlighted in green. Any arbitrary point $\boldsymbol{c}$ on this highlighted region can be characterized by the variable $l \in \mathbb{R}$, where $|l|$ is the distance away from the frame L_i, and $sgn(l)$ denotes which side of the link the point is on. If d is the width of the link, then

$$\boldsymbol{c} = \begin{bmatrix} |l| & \frac{d}{2} sgn(l) \end{bmatrix}^T. \tag{6.10}$$

The normal velocity of the point $\boldsymbol{c}(l)$, can be written explicitly as a function of l. If $\boldsymbol{v}^b_{\mathrm{S,L}_i}(t) = [v_x, v_y, \omega_z]^T$, then

$$\dot{c}_n(t) = sgn(l)(v_y + |l|\omega_z). \tag{6.11}$$

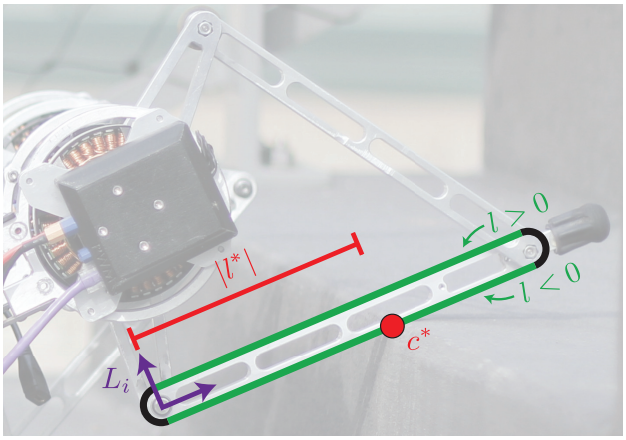


Figure 6.3: The portion of the link that is expected to make contact (B) is shown in green. Any point in this one dimensional set can be characterized by the parameter l. Using the velocity-based method, the robot can find l^*, which characterizes the true contact point.

At time t ($t_0 > t > t_f$), the values of l that satisfies the constraint in Eq. (6.8) is,

$$\dot{c}_n(t) = 0 \Rightarrow l = \pm v_y/\omega_z \tag{6.12}$$

The set of possible contact points, C can then be found by mapping these values of l to points using (6.10).

For $t_0 < t < t_f$, the set C contains two possible contact points. However, at time t_0, the candidate contact points also need to satisfy $\dot{c}_n(t_0^-) > 0$, due to the constraint in Eq. (6.9). Given that $l = \pm v_y/\omega_z$ for the two points in C and Eq. (6.11), one of the points in C violates the constraint in Eq. (6.9) and is removed from C. This results in a unique point at which contact could have occurred.

6.5 Implementation Results

6.5.1 Contact Localization in Simulation

In ideal conditions, the position, velocity, and torque based solutions all provide accurate contact localization. However, they differ in their sensitivity to noise. To evaluate this, we simulate the frictionless collision of a five-bar linkage with a point constraint. We assume that the collision link is fixed and known. We derive the kinematics and differential kinematics of the five-bar linkage as described in [18] and the dynamics using a constrained Lagrangian approach [90]. The dynamics are then projected onto the reduced coordinate space of the actuated joints. A constraint is added to this system for a frictionless point contact. This system is simulated in an event-driven framework [53] using `ode45` in MATLAB. The linkage is actuated with a Proportional-Derivative position controller computed at

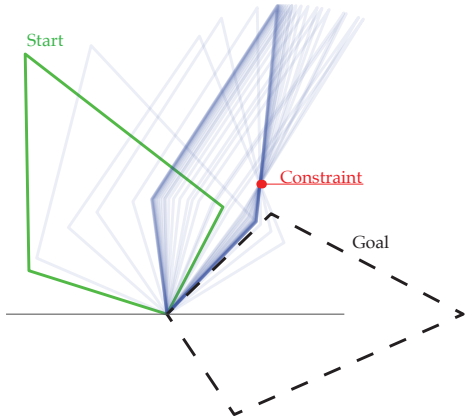


Figure 6.4: A sample trace of the simulation of the collision of a five-bar mechanism (one link is of length zero) with a point constraint(red) used to gauge the sensitivity of the discussed algorithms to model and process noise. The green trace is the initial state of the linkage which is commanded to align it's major axis with the x-axis. The darker blue trace is the resting position of the linkage after contact.

500 Hz outputting joint torques. The parameters for the simulation are shown in Table 6.1.

Using this simulation environment, we can test the sensitivity of the three algorithms described in this chapter to injected noise in the process parameters. The velocity and torque methods were tested twice: once with only encoder position as input and again with both encoder position and velocity as input. For the first case, labeled "Velocity($\boldsymbol{q}$)" and "Torque($\boldsymbol{q}$)", the velocity was computed by finite difference of the encoder position at the same dt (0.02 s) as the time window of the position-based method. In this case, the velocity method was indiscernible from the position method and are shown together. For the second case, labeled "Velocity($\boldsymbol{q}, \dot{\boldsymbol{q}}$)", and "Torque($\boldsymbol{q}, \dot{\boldsymbol{q}}$)", a velocity was computed at a higher frequency (4000 Hz) and then low pass filtered (3 dB cutoff frequency of 142 Hz.) The simulation executed nominally with a 16 bit encoder resolution as baseline noise.

For all sensitivity analyses, the performance is evaluated using the error over a 100 ms window starting 100 ms after contact is detected. Root-Mean-Squared (RMS) error over

Parameter	Value
Link Masses (m_1,m_2,m_3,m_4)	0.1 kg
Link Lengths (l_1, l_2, l_3, l_4, l_5)	0.08,0.15,0.15,0.08,0 m
Gravity	9.81 m s^{-2}
Contact Location	0.05 m, 0.09 m
Control Frequency	500 Hz
Position time window	0.02 s
Proportional Gain K_p	0.5 kg s^{-2}
Derivative Gain K_d	0.04 kg s^{-1}

Table 6.1: Parameters used for sensitivity analysis of the Contact Localization methods.

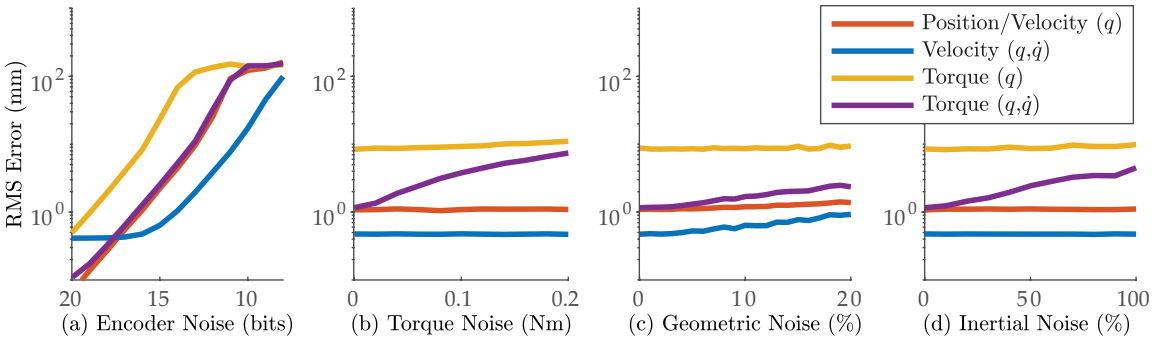


Figure 6.5: Effect of process and parametric noise on the accuracy of contact localization methods in simulation. All methods are highly sensitive to changes in position accuracy. The position and velocity contact localization outperform torque-based under all perturbations.

100 trials of each condition is reported in Fig. 6.5. The results described below are all statistically significant with a p value of less than 0.05. The z-statistic was used to compare the distribution of error computed at the extremes of the parameter sweep.

Sensitivity to Encoder noise

Gaussian noise was added to the generalized position coordinates from the simulation to emulate encoders with 8 to 20 bits of resolution with the standard deviation as 1 encoder count in radians. This proved to be the most important parameter tested, with all methods showing a large increase in RMS error with decreasing encoder resolution (Fig. 6.5a). The velocity-method with position and velocity input, Velocity($\boldsymbol{q}, \dot{\boldsymbol{q}}$), was the best method at low to moderate encoder resolutions, while all methods had sub-mm error for higher resolutions.

Sensitivity to Torque Noise

The torque based contact localization method requires the measurement of the generalized torques (joint torques in this case) to compute the external torques. Uniform torque noise of 0 to 0.2 N m was added to the measured torque. The torque method with position and velocity input, Torque($\boldsymbol{q}, \dot{\boldsymbol{q}}$), shows a significant increase in RMS error with increase in torque noise (while Torque($\boldsymbol{q}$) had uniformly higher error).

Sensitivity to Geometric Noise

Noise in measured dimensions are possible due to errors in design and manufacturing. We analyze the effect of up to 20% noise in the geometric parameters of the linkage. All methods except Torque($\boldsymbol{q}$) showed a statistically significant rise in error with increasing geometric noise, although the magnitude of the increase was less than for the other parameters tested. We attribute this to the weak signal to noise output of this method due too its dependence

on a clean velocity signal. A quick note: we uses the term *geometric noise* as a substitute for geometric uncertainty to keep the nomenclature consistent. The correct way to address noise in parameters of a model is with the term uncertainty.

Sensitivity to Inertial Noise

Careful calibration of geometric and inertial properties of each link and its contribution to the dynamic response can be tedious [59, 70]. To analyze the effect of errors in the dynamic robot model, we simulate up to 100 percent noise in the mass parameters of the linkage. As expected, both torque-based methods showed a statistically significant rise in error.

6.5.2 Contact Localization on the Minitaur Leg

The three contact localization methods were implemented and compared on a Minitaur robot. During contact localization, motor positions were measured using encoders mounted on the motors and velocities were estimated by taking the first order numerical differentiation of motor position. Motor torques were estimated from measured motor currents. For all three methods, it was assumed that contact occurred on the last link of the robot's leg.

A 100 ms time window was chosen for both the time window of the position-based method and the dt of the velocity estimate. The effects of varying this time window and the dt value are discussed below.

As discussed in Section 6.3.2, the torque-based method is dependent on an accurate dynamic model of the robot. This model was derived from link and motor mass distributions as measured from CAD models and actual link weights. Aerodynamic effects and frictional forces were ignored. Motor constants vary between individual motors and as such they were calibrated separately for each motor. Furthermore, as mentioned in Section 6.3.2, it is necessary to make a frictionless contact assumption to find a single solution.

Unfortunately, we were unable to obtain sensible results using the torque-based method. While the estimated external torque values (τ_{ext}) from the momentum-observer seemed to be realistic, the contact locations were not. We believe that this is due to the high sensitivity of contact location to any noise in the τ_{ext} estimates. This difficulty in achieving an accurate torque-based estimate provides further motivation for the method developed in this work. To compare the velocity-based method and position-based method, one of the robot's legs was swept into a stationary object and the two methods were used to estimate the contact locations. These estimated contact locations were then compared to ground truth contact location measurements. To get these accurate ground truth measurements, the obstacle was rigidly attached to the body of the robot, and the dimensions of the rig connecting the obstacle to the robot were measured. This experiment provides a scenario similar to a legged robot on stairs (Fig. 6.1), but provides ground truth contact locations, which was used to evaluate contact location estimates, within 1 cm of the true contact location. The experiment setup is shown if Fig. 6.6. Six different contact positions were used. At

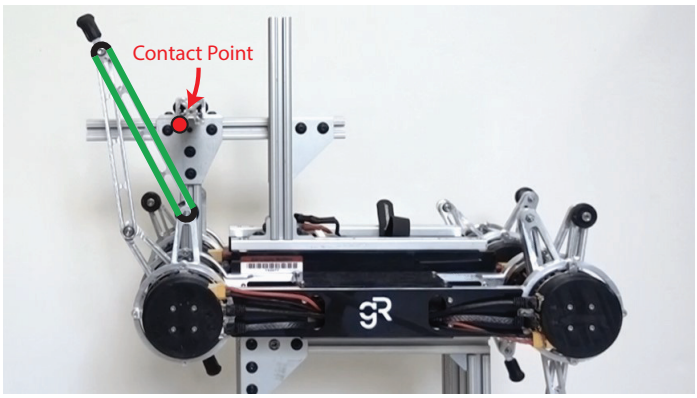


Figure 6.6: An experiment localizing contact on a Minitaur robot.

each position, 100 estimations of the contact location were made using the three different methods. The actual contact locations are shown as the dots in Fig. 6.7 and the estimated contact locations from the position-based, and velocity-based methods are shown as the crosses.

With a 100 ms time window for the position-based method and dt for the velocity estimate, the accuracy of the position-based and velocity-based methods was comparable. The position-based method had an average error of 0.57 cm, with a variance of 0.14 cm. The velocity-based method had an average error of 0.48 cm, with a variance of 0.12 cm.

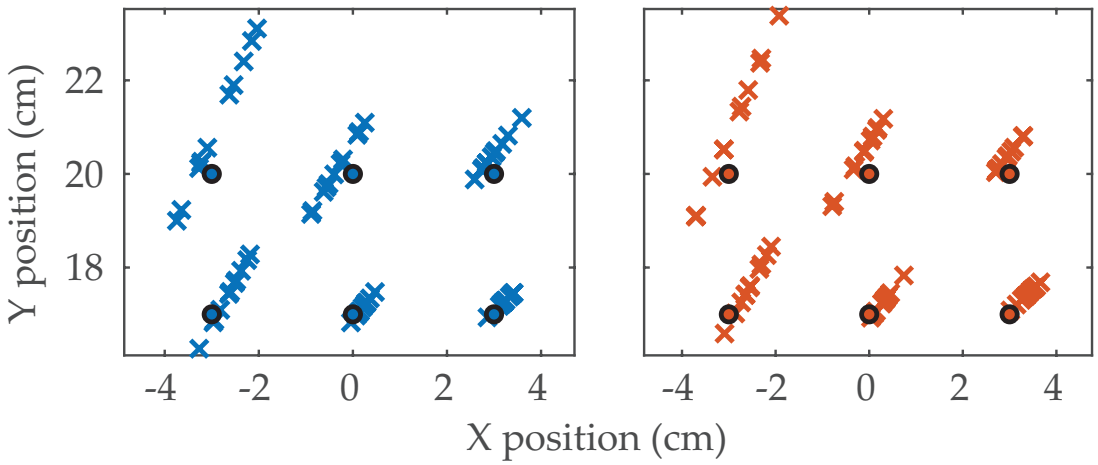


Figure 6.7: Estimated contact locations from the Minitaur experiment using the velocity-based (left) and the position-based (right) methods. The circles represent the actual contact locations, and the crosses represent the estimations made using either method. With the same time window, both methods result in similar contact location estimates.

The time window of the position-based method and the dt of the velocity estimates were simultaneously varied from 10 ms to 100 ms to analyze their effects on the contact location estimates. The results of this experiment is shown in Fig. 6.8. As expected, with a shorter time window and dt, both method resulted in higher error. Throughout the range of tested

time windows and dt values, both methods of contact localization had indistinguishable error. However, as shown in Section 6.5.1, the velocity-based method could have performed better than the position-based method if other methods of velocity estimation were used.

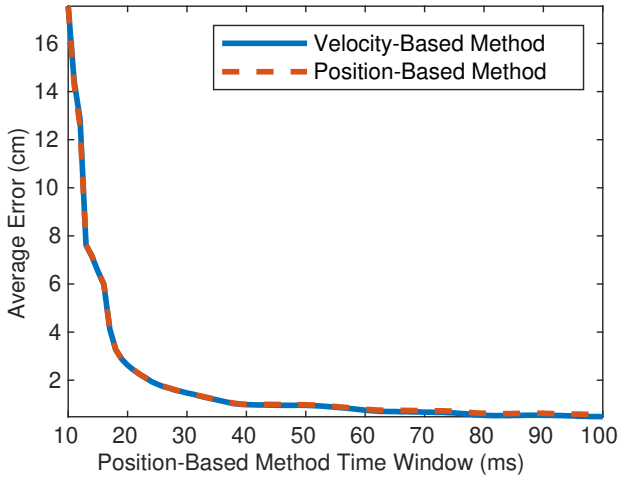


Figure 6.8: Average error of estimated contact point position from the Minitaur experiment using both the position-based method and the velocity-based method with a varying time window and velocity estimate dt size.

6.5.3 Contact Localization on the DD Hand

We demonstrate the velocity-based method with a non-point contact on the DDHand. One of the fingers is forced into an unexpected collision with a cylinder in a fixed orientation. The algorithm described in this chapter is employed to continuously track the estimated contact location. Fig. 6.9 shows the evolution of the linkage during the experiment along with the estimated contact points overlaid. The estimated contact location is seen tracking the curved surface of the cylinder. The small deviation at the start can be attributed to the lag due to the velocity filter running on board the DD Hand.

6.6 Discussion

In this chapter, we propose a velocity-based method of contact localization. Using velocity constraints, this method provides a codimension 1 set of possible contact points. The advantage of this method over previous methods is that it provides an instantaneous estimate of contact location, does not require a dynamical model of the robot, and only requires robot position and velocity measurements. We validate the performance of this method for planar robots both in simulation and the real world. In the plane, we show that kinematic methods of contact localization are superior to dynamic methods in the presence of noise.

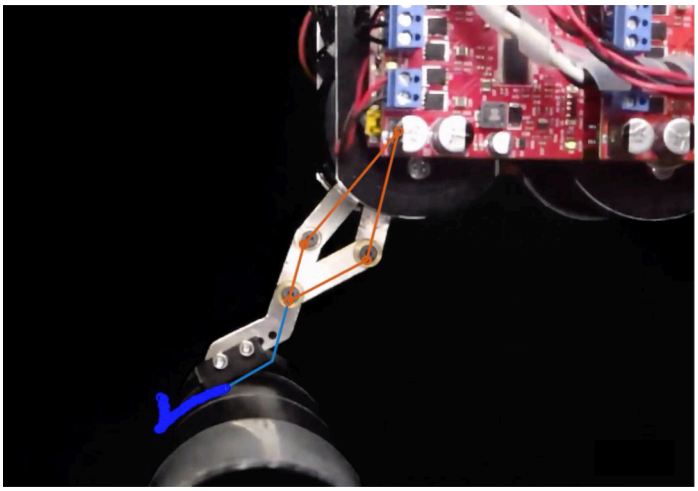


Figure 6.9: A snapshot from the experiment to detect contact of the DDHand with a cylindrical object.

Note that the velocity-based method is equivalent to a position-based method when the velocity is computed as a first order numerical differentiation over the same time window. For small time windows, the velocity and position-based methods produce similar results. This is shown in both the simulation and real-world experiments in Fig. 6.5 and Fig. 6.8, respectively. However, one advantage of the velocity-based method over the position-based method is that the accuracy of the velocity method improves when a better velocity estimate is provided. This can be achieved with either better numerical differentiation of position data [19] or by fusing other sensor data like acceleration [137].

For spatial robots, this contact localization method results in a one dimensional set of possible contact points leading to ambiguity in the location. To reduce this, a particle filter with a stationary contact point assumption is proposed in [129].

In the future, improvements to state-estimation techniques could allow for more accurate velocity estimates, and thus contact location estimates. Also, development of more intelligent particle filter motion models could extend the method to localizing non-stationary contact points on spatial robots.

Chapter 7

Applications of the DDHand in Small Parts Assembly

Introduction

Dexterous manipulation research has been advancing for forty years, yet still seems far from practical use. Some recent developments in hand speed and behavior might have brought us to the brink of practical application, even in demanding industrial assembly tasks. The goal of this project is to develop a dexterous manipulation capability with the breadth of scope, precision, robustness, and speed sufficient for industrial automation.

The key elements of the approach are:

1. Direct-drive hand. The use of Direct-drive in manipulator design is not new, but has only recently become practical for fingers.
2. 4DOF "intermittently planar" kinematics. Considerably simpler than most previous dexterous systems, the hand kinematics comprises two 2DOF fingers in a co-planar arrangement. We hypothesize that almost every required 3D manipulation can be mapped to a planar manipulation by placing the operating plane correctly.
3. Pre-grasp whole-world manipulation. Putting the dexterity before the grasp, or as part of the grasp, means the support surface is available as a third finger. The drawback is the additional time required. We hypothesize that the system will be fast enough to perform the manipulation and grasp in just a few seconds, fast enough for industrial overall cycle times as short as six seconds.
4. Non-prehensile primitives—pushing, squeezing, toppling, and pivoting—combined when possible with positive location, designed for open-loop precision and robustness.

To test the approach we focus on the assembly of an automobile door-latch subassembly (Fig. 7.1). Three small parts are to be assembled into a housing. Many stable poses of the parts are not directly available for an assemblable grasp, thus dexterity is required. This

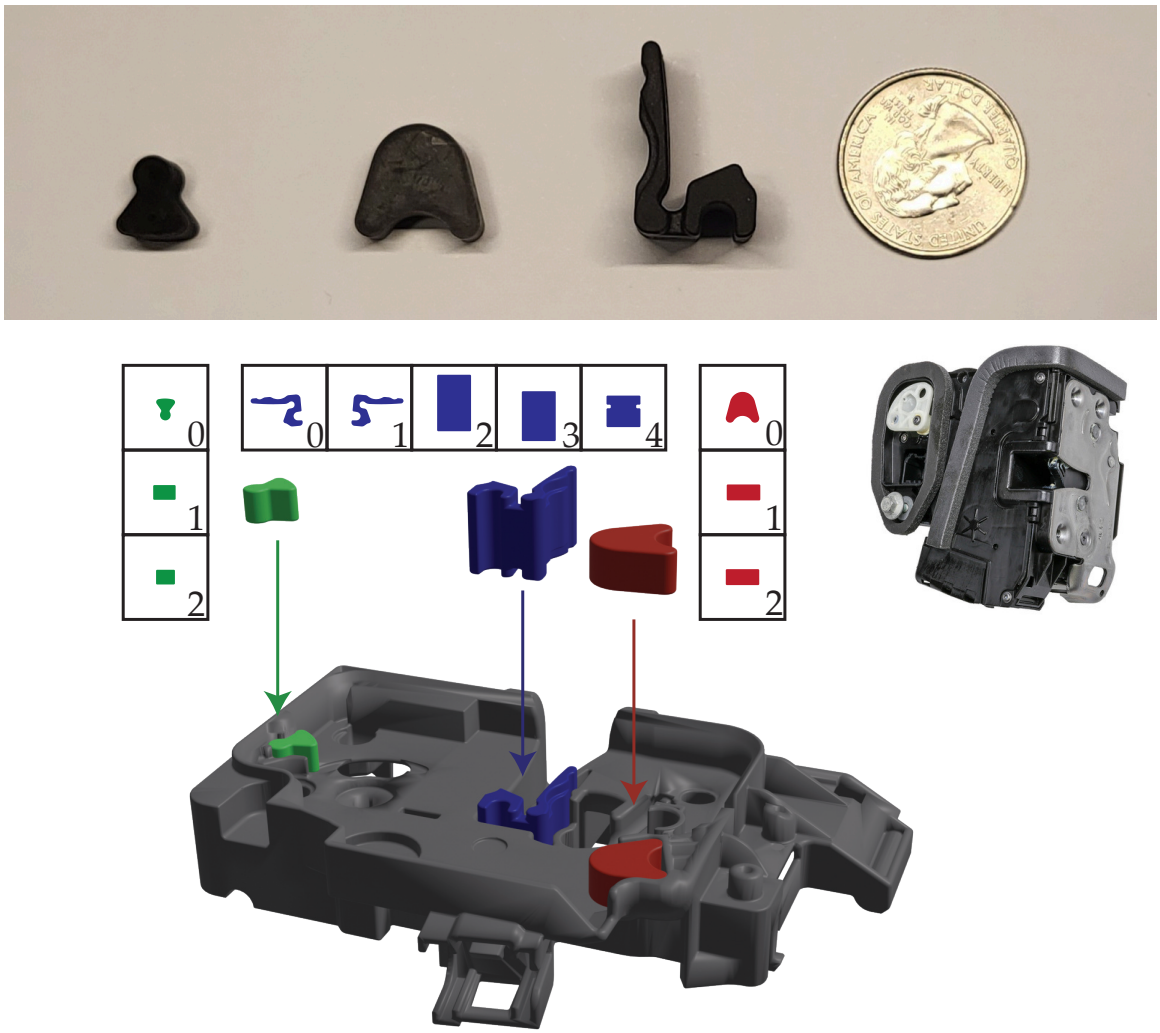


Figure 7.1: Top: (left to right) Small Bumper, Medium Bumper, Large Bumper, and a US quarter for scale. Bottom: The assembly task discussed in this work. The part shown on the top right is an automotive door latch. Three rubber parts need to be inserted into the housing (black): small bumper (green), medium bumper (red) and long bumper (blue.) The top-view of the stable poses is shown in the figure. Pose 0 for all objects is considered as the assemblable pose.

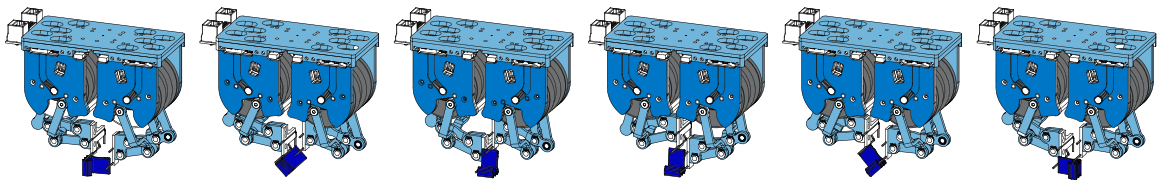


Figure 7.2: The Dexterous Direct-drive hand reorienting a long bumper to be assembled in an automotive door latch subassembly.

chapter presents an automation system and pipeline for this dexterous assembly task. The parts are presented singulated on a light table, in any stable orientation. The presenting pose is reported by a vision system, our system plans a sequence of motions to grasp the part in the assemblable pose, and the resulting plan is executed by a dexterous direct drive hand. We reports the results of over 400 trials of the system—at least 30 trials for each presenting pose of each part. The success rate varies from 83% for some part poses to 100% for others. We show that the speed is one advantage of this system — the required execution time was four seconds or less for two of the parts, but ran up to sixteen seconds for the third part. However, these times do not include the time for vision due to variations in some primitives' outcomes. To further speed up the process, we suggest to use existing primitives in the system to reduce object pose uncertainty, as presented in the discussion section.

7.1 Related Work

The simplest form of manipulation is pick-and-place, where the hand is only used for gripping, and the object moves only when rigidly gripped by the hand. When a required grasp pose is inaccessible as presented, a *regrasp* aka *place-and-pick regrasp* might work. The object might be picked and placed in an intermediate pose, then picked again in the required pose. Examples of the place-and-pick regrasp are seen in the Instant Insanity demo [96] and the Handey System [77, 122]. Unfortunately the pick-and-place regrasp tends to be slow, requiring multiple grasps and large arm motions.

The "dexterous hand" approach, first described in Salisbury's PhD thesis [111], uses three fingers, each with 3 DOFs, to perform in-hand manipulation of a grasped object. A considerable body of work followed Salisbury's lead and variations [93]. test

Manipulation before the grasp or as part of the grasp, such as pushing and squeezing, can be observed in early blocks-world work, and was the subject of Mason's PhD thesis in 1982 [83]. Since that time there have been numerous studies of pushing, squeezing, levering up, and related behaviors [8, 20, 22, 35, 47, 50, 79, 102].

Hand design is central to dexterous manipulation, including the trade-offs between simplicity and complexity. Salisbury's *dexterous hand* had three fingers, three motors per finger, all actuated. Others have employed anthropomorphic grippers with even more. At the other end of the spectrum, simple hands are typically focused on gripping [28, 85], but it is possible to use very simple hands for dexterous manipulation[21]. The tradeoff is that a simpler mechanism might require more complex behaviors. With the DDHand, we attempt to move away from the simplest extreme in hand design, and also away from the most complex extreme in behaviors. The simpler behaviors follow recent work on extrinsic dexterity [21] and shared grasping [48], extending the definition of the hand to include environmental constraints and gravity.

In Chapter 4, we describe the design of the Dexterous Direct-drive Hand (Dexterous DDHand), which is based on the earlier Direct-drive Hand (DDHand) [11]. Direct-drive actuation has been explored for manipulators for decades [5], but use of direct-drive in hands is a recent development. The speed and natural compliance of the Dexterous DDHand is essential, allowing compliant manipulation to occur prior to the grasp, or as part of the grasp, without blowing the time budget.

Dexterous behaviors can be complex. Division of these complex behaviors into smaller repeating blocks or *primitives* is common in robotics. Motion primitives can be hand-designed[24, 98], generated through motion planning [23], learned from demonstrations [94] or through reinforcement learning. These motion primitives can be composed into behaviors either manually, or with some task-level planning frameworks.

7.2 Door Latch Assembly Task

This chapter deals with the challenge of acquisition of parts when an immediate assemblable grasp is not available. The scenario is as follows. Consider the peg-in-hole assembly of non-uniform pegs which have been singulated on a surface (*e.g.* a table or conveyer belt.) The robot hand is required to acquire the pegs in a specific assemblable pose from the surface and assemble the objects. Sometimes, the grasp pose is not immediately feasible, it might place the hand in collision with the ground or other environmental constraints; or the grasp pose may be out of the workspace of the manipulator.

Specifically, we are concerned with the task of assembling three components into a housing which forms a subassembly for an automotive door latch as shown in Fig. 7.1. These components were chosen to represent a practical use case in industry as well as for their variations in shape and material. The small bumper and the medium bumper are made from high stiffness rubber while the large bumper is made from a more pliable low stiffness rubber. The small and medium bumpers have three unique stable poses while the long bumper has five. The top views of these poses are inset in Fig. 7.1.

Note that in order to simplify the analysis, we assume there is only one stable pose with a valid reachable grasp (pose 0 in Fig. 7.1). For the particular combination of objects and finger designs, this assumption may not hold true, meaning that our results can be improved just by choosing the most easily attained grasp pose.

7.3 Design of the Robotic Work Cell

To study the acquisition task, we develop a robotic work cell which includes an industrial manipulator with an end-effector for manipulation (Fig. 4.2), and a vision system for pose estimation of the parts. The schematic of this work cell is shown in Fig. 7.3. We describe the end-effector, fingertip design and supporting infrastructure in this section.

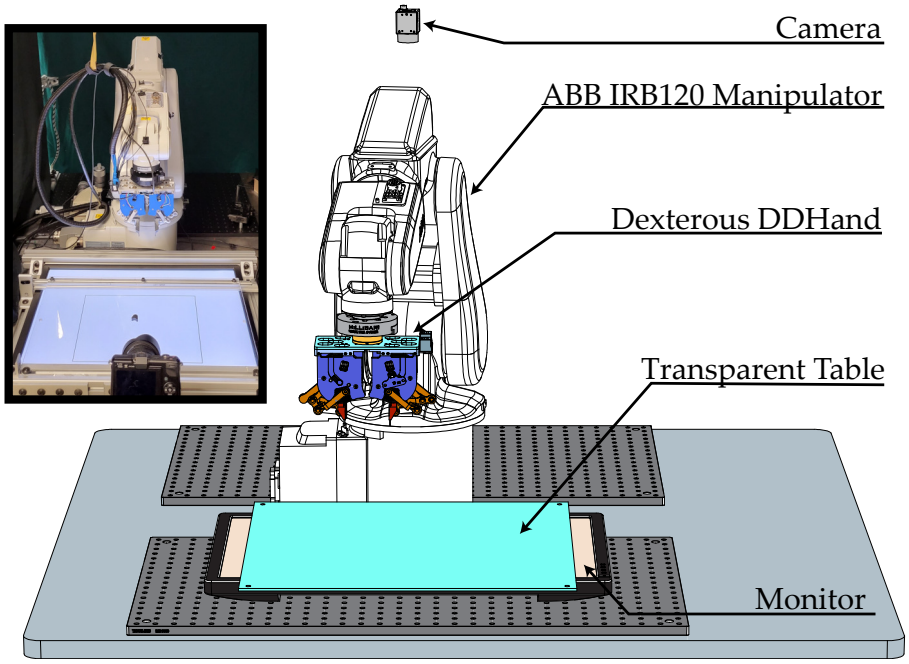


Figure 7.3: A schematic of the work cell setup for the acquisition experiments. A picture of the actual setup is inset on the top left.

7.3.1 End-effector Design

The task-level requirements of the end-effector (or hand; used interchangeably) decide its design direction:

1. Hand Payload: The maximum dimension for any of the parts in our case study is within 2 inches. The weight of the parts is on the order of tens of grams.
2. Arm payload: We assume a maximum payload of 5kg for the robot which includes the weight of the end-effector and the part it is grasping.
3. Dexterity: The hand needs sufficient range of motion to execute the motion primitives. A hand needs to be dexterous enough to accomplish these regrasp actions.
4. Cycle Time: grasping and reorienting steps need to be quick. There are cycle time restrictions on the assembly operation.
5. Robustness: At a higher level, the manipulation actions should be repeatable with high success rates. At the hardware level, we seek low maintenance throughout the lifetime of the system.

Low payload, faster cycle time and robustness requirements make direct-drive actuation a good candidate for this task. A direct-drive hand would enable high speed and transparency (transmission of force and motion between the end effector and the joints.), with the only caveat being reduced torque density. The Direct-Drive Hand (DDHand) [11] is one such end-effector system developed at the Manipulation Lab. The design was upgraded for

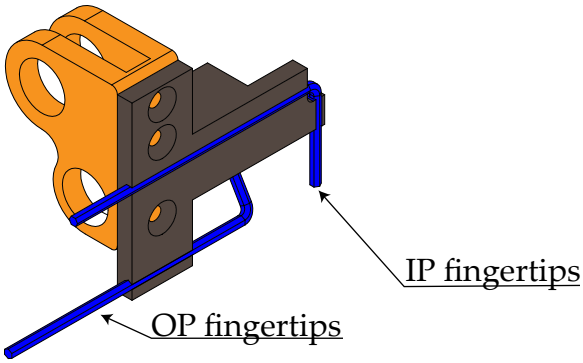


Figure 7.4: Fingertips for the Dexterous DDHand used for the regrasping task.

this task (Dexterous DDHand).

The hand operates in one plane at a time. In order to manipulate objects in arbitrary dimensions, the operating plane needs to be repositioned by a supporting arm. Moving the operating plane comes with a cost as arm motions are slower than finger motions due to larger inertias and higher reduction ratios in the arm.

7.3.2 Fingertip Design

Fingertip concepts along two exploration directions were considered: type of finger contact patch and the angle of the finger with respect to the operating plane of the DDHand. Point, plane and line contact patches were explored at 0, 45 and 90 degrees with respect to the operating plane of the hand.

Two fingertips were chosen: In-plane (IP) fingers – a line contact patch parallel to the operating plane; Out-of-plane fingers – a line contact patch orthogonal to the operating plane. The fingertips were situated offset from the center of the hand to allow for manipulation when the operating plane was parallel to the ground without collision. The fingertips were fabricated using 0.05 inch Allen wrenches mounted on a 3D printed adapter for the DDHand fingertip mounting system (Fig. 7.4).

7.3.3 Infrastructure

The Dexterous DDHand is integrated into a system including vision, motion planning and control subsystems, and inter-process communication to link the subsystems together.

Fig. 7.3 shows a schematic of the system hardware. There are three main elements: The Dexterous DDHand; the vision system including a light source, a table and a camera; and the industrial manipulator for hand positioning. The hand is mounted on a force-torque sensor which connects to a manual tool changer. The effector is positioned in 6 DOF space using an ABB IRB 120 manipulator.

A PointGrey Grasshopper color camera is used in the vision system, pointed at a computer monitor which provides a high contrast white backlight against the black parts. The

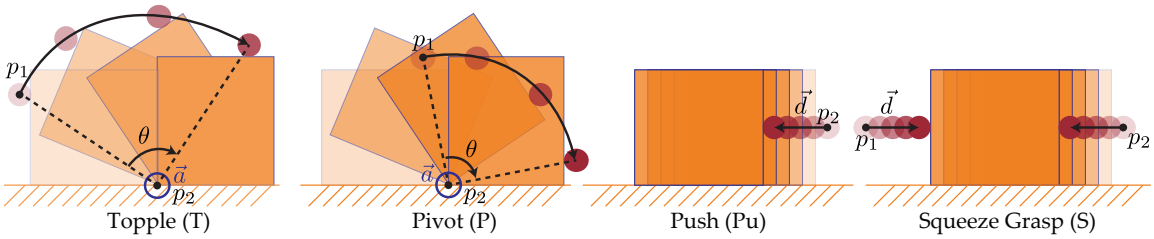


Figure 7.5: Motion primitives that make up the overall behavior of the system. Pivot and topple are variations of the same motion primitive and differ only in the location of the fingers.

vision system estimates the current pose of a part with a combination of classical vision techniques (rectangle detection and SIFT-based feature detectors) to locate the part in the camera frame which is calibrated to the robot. The vision system is written in python and runs on a dedicated 7th gen Intel NUC i7 on a per part basis, that is, only templates for one part are compared at a time to minimize the processing overhead.

7.4 Dexterous Behaviors

The system is manually programmed with a hierarchy of motion primitives and behaviors. Once the initial pose of the part is identified, a behavior lookup identifies the appropriate sequence and specification of motion primitive to execute in order to progress to the final assembly pose of the object.

We define four motion primitives (Fig. 7.5) for composition into behaviors: Pivot, Topple, Push, and Squeeze Grasp. These primitives are hand-designed sub-routines; they each take data in the form of 3-dimensional points and vectors, poses, or angle measures as parameters for the primitive exactly how to move with respect to the object being operated on.

Every one of the following primitives takes as input the object position and orientation as a 3-dimensional pose in the robot's frame. In the scope of our experiments, this input is provided by a 2D vision system but other systems can be used to provide the initial pose, e.g. the pose can be computed by forward propagation of the previous action.

Each primitive takes a set of geometric quantities as parameters and executes a trajectory in the task space of the DDHand.

Pivot (P) and Topple (T)

The pivot primitive rotates a part by some angle, about an edge in contact with the ground plane. A topple has the same effect as a pivot. The fingertip moves horizontally, making contact near the top of the object and continuing until the rotation of the object is complete. To make this action a little more robust, the fingertip motion is offset toward the center of rotation by a small amount to increase the normal force at the pivot point. With this

empirical adjustment, the point of contact on the ground does not slip nominally, and the motion ends with the fingertip pinning the object to the ground.

Both Pivot (P) and Topple (T) use the initial contact point of the toppling finger as a 3D point in the object frame shown as p_1 in Fig. 7.5. The pivot point p_2 marks the center of rotation of the finger. This input is tuned to match the point in space at which the object will remain in contact with the ground through the course of the Pivot (P) or Topple (T). We give a vector $\boldsymbol{a}$ to the pivot and topple primitives, where the magnitude of the vector indicates the angle θ in radians of the arc of the trajectory made by these primitives, and the direction $\hat{a}$ indicates the axis of rotation.

We derive a primitive that combines the pivot and the topple into another primitive called Pivot-Topple. This primitive results in an improvement in stability of the rotation.

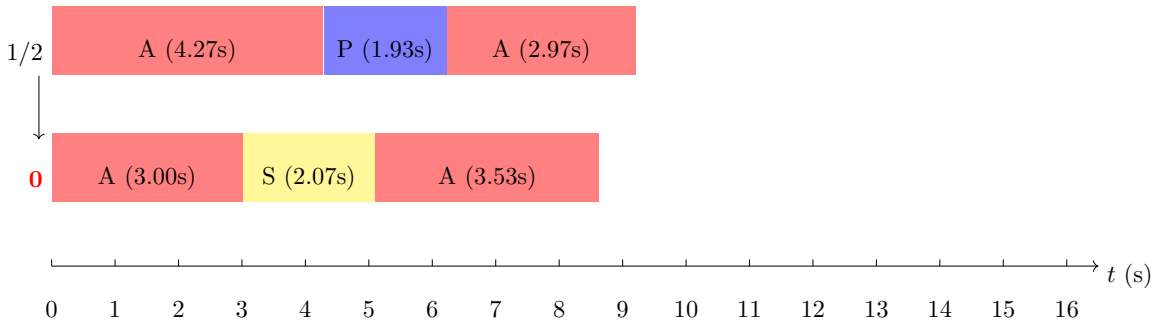
The pivot-topple primitive is hand-designed for this task. The control problem for pivoting has been extensively researched ([45, 47, 48]) and a suitably robust solution may be switched out for the one implemented in this work.

Push (P)

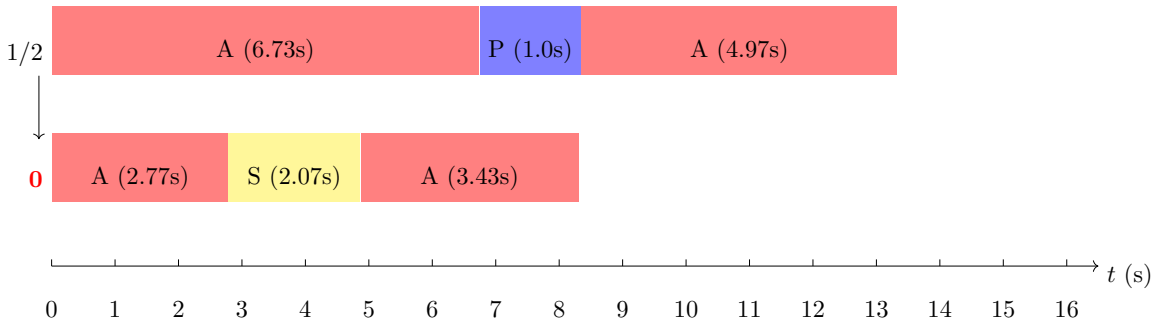
One finger moves horizontally through the objects initial position. A push can be used to eliminate uncertainty, aligning the part with the fingertip. The push uses the initial contact point of the pushing finger as a 3D pose relative to the object pose to mark where the pushing finger should make contact with the object at the start of the action, shown in Fig. 7.5 as p_2. The push primitive also uses a vector $\boldsymbol{d}$ as input, where the magnitude of the vector $|\boldsymbol{d}|$ indicates the length of the trajectory, and the normalized vector $\hat{d}$ indicates the direction in which to push.

Squeeze Grasp (S)

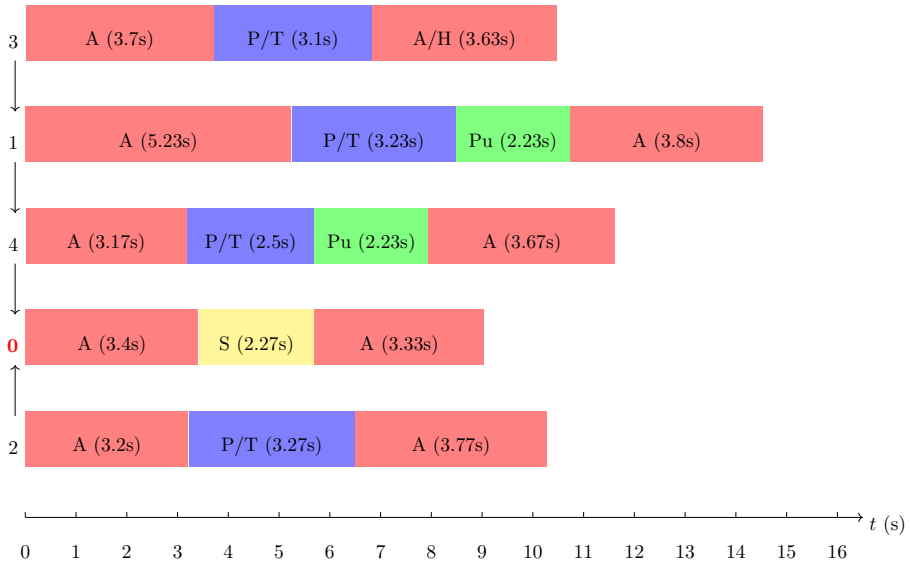
Two fingertips approach the object from opposite directions and apply a squeezing force. This primitive can eliminate some uncertainty, centering and/or aligning the object. The Squeeze Grasp uses two 3D poses p_1 and p_2 to indicate the initial contact point of finger 1 and finger 2, respectively. These points indicate where the fingers should begin the squeeze. A second input is given as a binary. This input indicates to the primitive whether the squeeze should be made going "in" or "out". For example, when grabbing an object with an inside squeeze, we use "in" with this primitive. Lastly, the length of the distance that each finger should move is given as a magnitude $|\overrightarrow{d}|$. The squeezing force is determined by the distance the endpoint of the squeeze penetrates into the object and the proportional gain of the DDHand joint controller mapped to the fingertip.



(a) Small bumper.



(b) Medium bumper.



(c) Long bumper.

Figure 7.6: Time spent in the transition between stable poses. For the assemblable pose (0) the grasp time is included in the total time. Here A denotes the time spent in reorienting the gripper and includes the time the arm takes to clear the vision system for measurement.

Part	Pose	Success	Success Rate	Primitive Time
Small Bumper	0	55/60	91.66%	2.07
	1	30/30	100%	4.0
	2	25/30	83.33%	4.0
Medium Bumper	0	30/30	100%	2.07
	1	30/30	100%	3.07
	2	30/30	100%	3.07
Long Bumper	0	82/90	91.11%	2.27
	1	30/30	100%	12.46
	2	27/30	90%	5.54 or 15.73
	3	25/30	83.33%	5.37 or 15.56
	4	30/30	100%	7.0

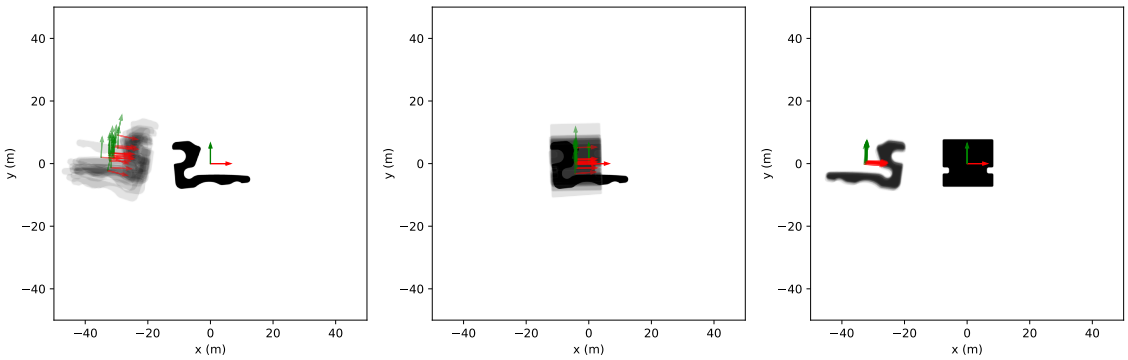
Table 7.1: Overview of the success rates and time taken to execute each manipulation. Note three points: 1) This is the aggregate time to execute all primitives required to reposition the part into its assemblable pose from the initial pose; 2) pose 2 and pose 3 of the long bumper each has two values for primitive time due to the uncertainty in the long bumper's orientation, which leads to different sequences of primitives used. 3) The vision system is called at each step and the vision processing is not included in this time – these times can be further reduced by removing the vision step.)

7.5 Experiments

The system was tested on the small, medium and long bumpers. We manually designed behaviors to start from each stable pose of the objects and end at the next pose as shown in Fig. 7.6. Fig. 7.2 shows a sequence of motions for pivoting the long bumper from pose 1 to 0. For each stable pose, we conduced at least 30 trials over which the success rate was measured. Success was declared when the part was grasped in the hand after the final squeeze-grasp primitive in pose 0. Failure was declared if, at any point, one of the primitives failed or the part was not successfully acquired in the hand. Failures due to errors in the vision system are not reported. These were considered reset events. At least 30 trials of each stable pose were executed with the part starting in a random location on a 8x11 rectangular light table. Table 7.1 shows the results of these experiments.

All three objects have a common mode of failure in the squeeze grasp that is executed by the IP fingers. Due to errors in the reported pose from the vision system and the cantilevered design of the fingertips, there is a possibility that the squeeze grasp doesn't succeed. This showed up in all the failures when the objects were in pose 0.

The second failure mode was seen when starting from any pose that shows up as a degenerate rectangle in the top view. For the small and medium bumpers, pose 1 and 2 look very similar in the top view and the vision system was unable to distinguish between them. To handle this, the behavior for both pose 1 and 2 was implemented as the same behavior. This resulted in different pivoting forces for both the poses resulting in a lower success probability for one of them. The medium bumper did not show this behavior as it



(a) (b) (c)

Figure 7.7: A plot of the uncertainty in the Long Bumper's final pose after another Pivot, with an additional uncertainty reduction Push. The uncertainty is eliminated entirely. The center of the plot shows the Long Bumper's initial pose, and silhouettes show the final pose of the Long Bumper with 30 trials.

has a length to width ratio that is close to unity.

This issue of reporting inconsistent stable poses by the vision system also showed up in the reorientation of the long bumper. Consider the long bumper in pose 2 or 3; depending on the orientation of the rectangle, a pivot-topple can result in either pose 0 or pose 1. For this reason, we report two durations for the execution of the long bumper. The longer durations are the worst case scenario when the manipulation of pose 2 or 3 results in pose 1 and the shorter durations are the best case scenarios when the result is pose 0. For ease of reporting, we show the best case execution for pose 2 and the worst case execution for pose 3 in Fig. 7.6.

In the worst case, the chain of stable poses that the long bumper goes through is pose 2/3 $\rightarrow$ pose 1 $\rightarrow$ pose 4 $\rightarrow$ pose 0. This increases the probability of failure which is why most failures were seen when the initial stable pose is either pose 2 or 3.

7.6 Discussion

In this chapter we have shown the utility of extrinsic dexterity in a real-world application. By pre-grasping with a dexterous and agile end-effector, we show an alternative to place-and-pick regrasping actions.

In addition, our prototype system solves an industrially relevant problem which is extendible through state-of-the-art approaches in the field of robotic manipulation. We discuss three such extensions in this section.

7.6.1 Ease of Redeployment

Making a case for automation is challenging in manufacturing. Two issues govern the economics of deploying automation: high throughput requirements in favor of, and constant

refreshes against. The popular teach-and-repeat method requires a laborious reprogramming, making automation systems less economical. The methods described in this chapter take a different approach. Instead of programming teach points in the task space of the end-effector, the method provides generalized primitives parameterized by geometric quantities. Reconfiguring the system for a new part requires only geometric knowledge of the part to inform the parameterization of the primitives and their order of execution.

This method can also benefit from manipulation research in automatic generation of primitives given a task description, including trajectory optimization [117], sampling [23] and learning-based [71] methods.

7.6.2 Robust execution of Motion Primitives

At the primitive level, the work implements hand-designed trajectories parameterized by geometric quantities. These are executed on the system with an impedance control scheme using a proportional-derivative controller to simulate a spring-damper system. This open-loop execution with compliance does not result in primitives robust to initial, sensor or process noise. This results in some of the failure modes for the reported experiments. The primitives can be further augmented with control policies that have been recently shown robust to noise in geometry and initial pose [46].

7.6.3 Uncertainty Reduction Motions

Ideally, executing a primitive on a component would reliably result in the target stable pose, but the exact object poses is often subject to uncertainty. Moreover, since multiple primitives must be sequenced together to get the object into its final assemblable stable pose, the errors could propagate and accumulate. Fig. 7.7a shows an example of this uncertainty stack-up of the long bumper after two pivot primitives to get from pose 2 to pose 0. Errors tend to compound with each action, eventually preventing additional primitives to be reliably executed without localizing the object.

In the current implementation, the vision system gives accurate object pose estimation after every primitive. Our vision system has high accuracy of estimating the exact pose of the object, but calling this vision system increases the Takt time as the robot needs to clear the workpiece to avoid occlusions. It is beneficial to minimize the calls to the vision system and query object stable pose only at the beginning and end of the reorientation operation.

Our preliminary experiments have shown that certain actions can help in eliminating pose uncertainties to a certain extent. Actions like pivot and topple increase the uncertainty of the final pose of the part, while actions like push and squeeze grasp reduce it [136]. Fig. 7.7b shows the propagation of uncertainty in the manipulation from pose 1 to 4 of the long bumper with a single push. The uncertainty collapses in one direction. Fig. 7.7a shows the effect of uncertainty from pose 4 to 0 with two pushes. The uncertainty is now completely collapsed.

Chapter 8

Future Work and Conclusion

8.1 Future Work

In this section, we first briefly cover extensions to work described in this thesis and then address the task planning and control with direct-drive end-effectors in detail.

8.1.1 Design of the DDHand

The designs of direct-drive hands (DDHand and Dexterous DDHand) are prototypes. They are not yet fit for deployment in industry and will benefit from a redesign from the perspective of industrial design and design-for-manufacturing principles.

There are a few design upgrades that would improve the design and utility of the DDHand:

1. Brakes: Low continuous torque output from the motors, the DDHand does not have a high payload capacity. Direct-drive motors can operate at higher torques than the specified continuous torque for a short amount of time limited only by thermal properties of the winding. To apply high gripping forces, the motors could operate in their peak torque regime for a short amount of time and brakes at the motor joints could be engaged to maintain a preloaded force.
2. Orientation control of the gripper jaw: The DDHand and the Dexterous DDHand both lack the ability to control the orientation of the fingertip. The orientation of the fingertip is dependent on the position of the joints for the DDHand hand and is fixed perpendicular to the palm for the Dexterous DDHand. To make this controllable, link 8 (see Fig. 4.2 (right)) may be replaced with a prismatic joint.

8.1.2 Dynamic Behaviors

Along with improvements to the robustness of the move-until-touch and smack-and-snatch behaviors discussed in Chapter 5, there is scope for leveraging the benefits of direct-drive

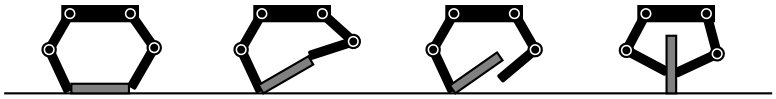


Figure 8.1: Levering up (also part of flip-and-pinch [23]) task with a two fingered gripper. The task has to be planned in a dynamic context as the object being manipulated is partially in freefall when a discrete contact switch is required.

actuation in other behaviors. Levering up (see Fig. 8.1), flip-and-pinch [23], throwing and catching [84], pizza peel manipulation [41], cloth unfolding [36] are examples of dynamic manipulation that could be explored with the DDHand or direct-drive actuation.

8.1.3 Contact Localization

Although the contact localization method described in Chapter 6 works well in the plane, for spatial robots, the contact localization method results in a one dimensional set of possible contact point. This leads to ambiguity in the location of contact. To reduce this ambiguity to a point, Wang et al. [129] propose the use of a particle filter with a stationary contact point assumption. In the future, improvements to state-estimation techniques could allow for more accurate velocity estimates, and thus contact location estimates. Development of more intelligent particle filter motion models could extend the method to localizing non-stationary contact points on spatial robots. Moreover, we envision the use of these localization methods for shape recovery using a Simultaneous Localization and Mapping context [68].

8.1.4 Task Planning and Control

Experiments in this thesis have been executed with manually planned trajectories. Automating the generation of motion plans and controls to accomplish a task with the DDHand is the perfect next research direction.

Most task planning approaches in manipulation use quasistatic assumptions to simplify computation. With the improved speed, bandwidth and transparency of direct-drive and quasi-direct-drive end-effectors, we have to drop these assumptions to extract the most performance. At higher speeds the dynamic feasibility of the generated trajectories needs to be considered; an area of research that has been well explored in ground and aerial locomotion [86, 116] and trajectory planning for manipulators [29]. Cheng et al. [23] show motion plan generation for the DDHand using a Contact-mode guided quasidynamic planner in 3D. The work also shows the open-loop execution of generated plans on the Dexterous DDHand system.

To further add robustness to these open-loop plans we look at two extensions: Hybrid Force-velocity control and Task-space impedance control.

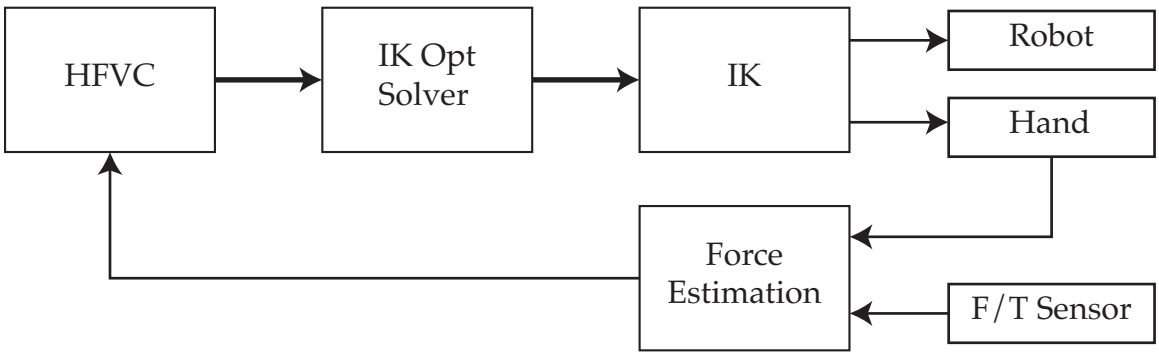


Figure 8.2: HFVC implementation pipeline on the DDHand.

Hybrid Force Velocity Control

Hybrid Force Velocity Control [46, 82] divides up the task space of a robot into orthogonal force and velocity control directions. Hou and Mason [46] have shown the efficacy of HFVC for robust execution of contact-rich tasks using a single point finger. There is merit to the idea of extending the method to two point contacts using the two fingers of the DDHand. Moreover, the improved speed, bandwidth and sensitivity of the DDHand allows us to modulate the force and velocity much faster than before.

To execute the output of a motion plan, the control architecture can be thought of as a *coarse/fine* controller [108] where the arm is executing the coarse motions of getting the hand's work space centered on the finger trajectories and the fingers are doing fine adjustments to locate the contacts appropriately.

The 10 DOF redundant configuration of the hand-robot system needs to be computed from two task space finger locations. Assuming point fingers, the inverse kinematics solution described in Chapter 4 decouples these 10 DOF into 6 DOFs of the finger tips and 4 DOFs of null motions. The 4 null space degrees of freedom can be solved by optimizing a cost function (IK opt solver in Section 8.1.4) that penalizes the motion of the robot more than the motion of the gripper while respecting joint limits and dynamics. The low inertia fingers can provide faster force measurements (transformed from motor torques) which can be fused with the measurements from a F/T sensor mounted on the wrist of the robot.

Once we have high-bandwidth position control of the two fingertips in cartesian space (x, y, z coordinates) we can implement a cartesian admittance control for the force control directions and a velocity controller in the orthogonal directions. Section 8.1.4 shows a simplified block diagram for the implementation of hybrid force-velocity control on the DDHand.

Task-space Impedance Control

An equivalent alternative to force-velocity control is to use task-space impedance control. Force control through this method is generally achieved by offsetting the setpoint to penetrate the surface on which force is to be applied. This method is briefly discussed in a manual plan generation context in Chapter 7.

8.2 Conclusion

Advances in motor design and increasing torque and power densities have ushered in a new era of direct-drive and quasi-direct-drive applications ranging from electric cars [1], to washing machines [103]. This thesis takes the first steps towards the use of direct-drive actuation in robotic end-effectors. In this thesis, we first identify transparency as a requirement for reactive interactions. We formally define transparency using the collision reflex metric and analytically compare systems with varying motor sizing and structural requirements. We show that the analysis favors direct-drive and quasi-direct-drive favorably. Next, we outline the design of the DDHand and the Dexterous DDHand – 4 DOF hands with parallel kinematics. We then compare the transparency of direct-drive (DDHand) and geared (Schunk Gripper) systems using a simple move-until-touch behavior and demonstrate a novel smack-and-snatch grasping behavior for the DDHand which is not possible without transparent actuation. We propose an algorithm for contact localization using velocity estimates with a requirement of high perception transparency and implement the method on a Ghost Robotics' Minitaur and the DDHand. Finally, we show preliminary results of the utility of the Dexterous DDHand in regrasping tasks in an industrial application. We conclude the thesis by outlining several improvements and research directions as next steps.

Bibliography

[1] First drive: Rivian delivers the electric truck we've been waiting for. URL https://social.techcrunch.com/2021/09/28/first-drive-rivian-delivers-the-electric-truck-weve-been-waiting-for/. 78

[2] Agilent. *Direct Drive Robot User Guide*. March 2014. URL http://www.agilent.com/cs/library/usermanuals/public/G5430-90003_R00_DDRUG_EN.pdf. 6

[3] D. Alba, M. Armada, and R. Ponticelli. *An Introductory Revision to Humanoid Robot Hands*, pages 701–712. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-22992-6 978-3-540-29461-0. doi: 10.1007/3-540-29461-9_69. 6

[4] H. Asada. Dynamic analysis and design of robot manipulators using inertia ellipsoids. In *Proceedings. 1984 IEEE International Conference on Robotics and Automation*, volume 1, pages 94–102, 1984. doi: 10.1109/ROBOT.1984.1087211. 14

[5] H. Asada, T. Kanade, and I. Takeyama. Control of a Direct-Drive Arm. *Journal of Dynamic Systems, Measurement, and Control*, 105(3):136, 1983. ISSN 00220434. doi: 10.1115/1.3140645. 2, 6, 21, 66

[6] Haruhiko Asada and Il Hwan Ro. A Linkage Design for Direct-Drive Robot Arms. *Journal of Mechanisms Transmissions and Automation in Design*, 107(4):536, 1985. ISSN 07380666. doi: 10.1115/1.3260760. 31

[7] Y. Asano, T. Kozuki, S. Ookubo, M. Kawamura, S. Nakashima, T. Katayama, I. Yanokura, T. Hirose, K. Kawaharazuka, S. Makino, Y. Kakiuchi, K. Okada, and M. Inaba. Human mimetic musculoskeletal humanoid Kengoro toward real world physically interactive actions. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 876–883, November 2016. doi: 10.1109/HUMANOIDS.2016.7803376. 37

[8] Vincent Babin, David St-Onge, and Clément Gosselin. Stable and repeatable grasping of flat objects on hard surfaces using passive and epicyclic mechanisms. 55:1–10, 2019. 65

[9] VICTOR Barasuol, GEOFF Fink, MICHELE Focchi, DARWIN G Caldwell, and CLAUDIO Semini. On the detection and localization of shin collisions and reactive actions in quadruped robots. In *International Conference on Climbing and Walking Robots*, 2019. 49, 51

[10] Dominik Bauer, Cornelia Bauer, Arjun Lakshmipathy, Roberto Shu, and Nancy S. Pollard. Towards very low-cost iterative prototyping for fully printable dexterous soft robotic hands. In *International Conference on Soft Robotics*, pages 1–8, 2022. 21

[11] Ankit Bhatia, Aaron M. Johnson, and Matthew T. Mason. Direct drive hands: Force-

motion transparency in gripper design. In *Proceedings of Robotics: Science and Systems*, June 2019. 4, 10, 32, 35, 41, 44, 45, 50, 66, 67

[12] Ankit Bhatia, Aaron M. Johnson, and Matthew T Mason. Direct Drive End-Effectors with Parallel Kinematics, US Patent Application No. 17/326,719, May 2021. 4

[13] Antonio Bicchi. Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity. *IEEE Transactions on robotics and automation*, 16(6):652–662, 2000. ISSN 1042-296X. doi: 10.1109/70.897777. 6

[14] Antonio Bicchi, J. Kenneth Salisbury, and David L. Brock. Contact sensing from force measurements. *The International Journal of Robotics Research*, 12(3):249–262, 1993. doi: 10.1177/027836499301200304. 53

[15] Lionel Birglen and Clement M. Gosselin. Force Analysis of Connected Differential Mechanisms: Application to Grasping. *The International Journal of Robotics Research*, 25(10):1033–1046, 2006. 5

[16] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schaeffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, and G. Hirzinger. The KUKA-DLR Lightweight Robot arm - a new reference platform for robotics research and manufacturing. In *International Symposium on Robotics*, pages 1–8, June 2010. 5

[17] J. E. Bobrow, B. Martin, G. Sohl, E. C. Wang, F. C. Park, and Junggon Kim. Optimal robot motions for physical criteria. *Journal of Robotic Systems*, 18(12):785–795, December 2001. ISSN 0741-2223, 1097-4563. doi: 10.1002/rob.8116. 37

[18] G. Campion, Qi Wang, and V. Hayward. The Pantograph Mk-II: A haptic instrument. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 193–198, August 2005. doi: 10.1109/IROS.2005.1545066. 6, 30, 56

[19] Paul S Carpenter, Ronald H Brown, James A Heinen, and Susan C Schneider. On algorithms for velocity estimation using discrete position encoders. In *IEEE Industrial Electronics Conference*, volume 2, pages 844–849, 1995. 62

[20] Nikhil Chavan-Dafle and Alberto Rodriguez. Prehensile pushing: In-hand manipulation with push-primitives. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6215–6222. IEEE, 2015. 65

[21] Nikhil Chavan-Dafle, A. Rodriguez, R. Paolini, Bowei 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 *IEEE International Conference on Robotics and Automation*, pages 1578–1585, 2014. 65

[22] Xianyi Cheng, Eric Huang, Yifan Hou, and Matthew T. Mason. Contact mode guided sampling-based planning for quasistatic dexterous manipulation in 2D. In *IEEE International Conference on Robotics and Automation*, 2021. 65

[23] Xianyi Cheng, Eric Huang, Yifan Hou, and Matthew T Mason. Contact mode guided motion planning for quasidynamic dexterous manipulation in 3D. *arXiv:2105.14431 [cs.RO]*, 2021. 66, 74, 76

[24] Nikhil Chavan Dafle, Alberto Rodriguez, Robert Paolini, Bowei Tang, Siddhartha S. Srinivasa, Michael Erdmann, Matthew T. Mason, Ivan Lundberg, Harald Staab, and Thomas Fuhlbrigge. Extrinsic dexterity: In-hand manipulation with external forces. In *IEEE International Conference on Robotics and Automation*, pages 1578–1585, May 2014. doi: 10.1109/ICRA.2014.6907062. 66

[25] Ravinder S Dahiya, Philipp Mittendorfer, Maurizio Valle, Gordon Cheng, and Vladimir J Lumelsky. Directions toward effective utilization of tactile skin: A review. *IEEE Sensors Journal*, 13(11):4121–4138, 2013. 51

[26] Konstantinos Dermitzakis, Juan Pablo Carbajal, and James H. Marden. Scaling laws in robotics. *Procedia Computer Science*, 7:250–252, 2011. ISSN 1877-0509. doi: 10.1016/j.procs.2011.09.038. 21

[27] Aaron M Dollar and Robert D Howe. The highly adaptive SDM hand: Design and performance evaluation. *The international journal of robotics research*, 29(5):585–597, 2010. 5

[28] A.M. Dollar and R.D. Howe. A robust compliant grasper via shape deposition manufacturing. 11(2):154–161, April 2006. ISSN 1083-4435. 65

[29] Bruce Donald, Patrick Xavier, John Canny, and John Reif. Kinodynamic motion planning. *Journal of the ACM*, 40(5):1048–1066, 1993. 76

[30] M. Ebner and R.S. Wallace. A direct-drive hand: Design, modeling and control. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1668–1673, Nagoya, Japan, 1995. ISBN 978-0-7803-1965-3. doi: 10.1109/ROBOT.1995.525514. 6, 21

[31] Kevin C. Galloway, G. C. Haynes, B. Deniz Ilhan, Aaron M. Johnson, Ryan Knopf, Goran Lynch, Benjamin Plotnick, Mackenzie White, and D. E. Koditschek. X-RHex: A Highly Mobile Hexapedal Robot for Sensorimotor Tasks. Technical report, University of Pennsylvania, Philadelphia, PA, 2010. 6, 37

[32] David V. Gealy, Brent McKinley, Stephen Yi, Philipp Wu, Phillip R. Downey, Greg Balke, Allan Zhao, Menglong Guo, Rachel Thomasson, Anthony Sinclair, Peter Cuellar, Zoe McCarthy, and Pieter Abbeel. Quasi-direct drive for low-cost compliant robotic manipulation. In *IEEE International Conference on Robotics and Automation*. IEEE, 2019. 6

[33] Genesis. Live Drive. October 2017. URL http://www.genesis-robotics.com. 6

[34] R.C. Goertz. Manipulators used for handling radioactive materials. In Edward Bennett, James Degan, and Joseph Spiegel, editors, *Human Factors in Technology*, chapter 27, pages 425–443. McGraw-Hill, 1963. 9

[35] Kenneth Y. Goldberg. Orienting polygonal parts without sensors. 10:201–225, 1993. 65

[36] Huy Ha and Shuran Song. FlingBot: The Unreasonable Effectiveness of Dynamic Manipulation for Cloth Unfolding. In *Conference on Robot Learning*, pages 24–33. PMLR, January 2022. URL https://proceedings.mlr.press/v164/ha22a.html. 76

[37] S. Haddadin, N. Mansfeld, and A. Albu-Schäffer. Rigid vs. elastic actuation: Requirements performance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5097–5104, October 2012. doi: 10.1109/IROS.2012.6386227. 6, 21, 22

[38] Sami Haddadin. *Towards Safe Robots: Approaching Asimov's 1st Law*, volume 90. Springer, 2013. 9, 10

[39] Sami Haddadin, Alessandro De Luca, and Alin Albu-Schaffer. Robot Collisions: A Survey on Detection, Isolation, and Identification. *IEEE Transactions on Robotics*, 33(6):1292–1312, December 2017. ISSN 1552-3098, 1941-0468. doi: 10.1109/

TRO.2017.2723903. 53, 55

[40] Blake Hannaford. A design framework for teleoperators with kinesthetic feedback. *IEEE transactions on Robotics and Automation*, 5(4):426–434, 1989. 8

[41] Mitsuru Higashimori, Keisuke Utsumi, Yasutaka Omoto, and Makoto Kaneko. Dynamic Manipulation Inspired by the Handling of a Pizza Peel. *IEEE Transactions on Robotics*, 25(4):829–838, August 2009. ISSN 1941-0468. doi: 10.1109/TRO.2009.2017085. 76

[42] N. Hogan. Mechanical impedance control in assistive devices and manipulators. In *Joint Automatic Control Conference*, 1980. 5

[43] Neville Hogan. Impedance Control: An Approach to Manipulation. In *American Control Conference*, pages 304–313, June 1984. doi: 10.23919/ACC.1984.4788393. 8, 15

[44] Peter F. Hokayem and Mark W. Spong. Bilateral teleoperation: An historical survey. *Automatica*, 42(12):2035–2057, 2006. ISSN 0005-1098. URL http://www.sciencedirect.com/science/article/pii/S0005109806002871. 8

[45] Anne Holladay, Robert Paolini, and Matthew T. Mason. A General Framework for Open-Loop Pivoting. In *IEEE International Conference on Robotics and Automation*, 2015. 70

[46] Yifan Hou and Matthew T. Mason. Robust Execution of Contact-Rich Motion Plans by Hybrid Force-Velocity Control. *International Conference on Robotics and Automation*, pages 1933–1939, May 2019. doi: 10.1109/ICRA.2019.8794366. 74, 77

[47] Yifan Hou, Zhenzhong Jia, and Matthew T. Mason. Fast Planning for 3D Any-Pose-Reorienting Using Pivoting. In *IEEE International Conference on Robotics and Automation*, pages 1631–1638, May 2018. doi: 10.1109/ICRA.2018.8462834. 65, 70

[48] Yifan Hou, Zhenzhong Jia, and Matthew Mason. Manipulation with shared grasping. In *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, July 2020. doi: 10.15607/RSS.2020.XVI.086. 65, 70

[49] K. Hsiao, S. Chitta, M. Ciocarlie, and E. Gil Jones. Contact-reactive grasping of objects with partial shape information. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1228–1235, October 2010. doi: 10.1109/IROS.2010.5649494. 39

[50] Eric Huang, Ankit Bhatia, Byron Boots, and Matthew Mason. Exact bounds on the contact driven motion of a sliding object, with applications to robotic pulling. In *Proceedings of Robotics: Science and Systems*, 2017. 65

[51] Jonathan Hurst. Walk this way: To be useful around people, robots need to learn how to move like we do. *IEEE Spectrum*, 56(03):30–51, 2019. URL https://ieeexplore.ieee.org/abstract/document/8651932. 6

[52] Masayuki Inaba, Yukiko Hoshino, Kenichiro Nagasaka, Tetsuo Ninomiya, Satoshi Kagami, and Hirochika Inoue. A full-body tactile sensor suit using electrically conductive fabric and strings. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 450–457, 1996. 51

[53] Aaron M Johnson, Samuel A Burden, and Daniel E Koditschek. A hybrid systems model for simple manipulation and self-manipulation systems. *The International*

Journal of Robotics Research, 35(11):1354–1392, September 2016. ISSN 0278-3649. doi: 10.1177/0278364916639380. 56

[54] Lucas Joseph, Joshua K. Pickard, Vincent Padois, and David Daney. Online velocity constraint adaptation for safe and efficient human-robot workspace sharing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 11045–11051, October 2020. doi: 10.1109/IROS45743.2020.9340961. 9

[55] David Kahaner, Cleve Moler, and Stephen Nash. *Numerical Methods and Software*. Prentice-Hall, Inc., USA, 1989. ISBN 0-13-627258-4. 44

[56] Simon Kalouche. *Design for 3D Agility and Virtual Compliance Using Proprioceptive Force Control in Dynamic Legged Robots*. PhD thesis, , Carnegie Mellon University, Pittsburgh, PA, August 2016. 14

[57] M. Kaneko and K. Tanie. Contact point detection for grasping an unknown object using self-posture changeability. *IEEE Transactions on Robotics and Automation*, 10 (3):355–367, June 1994. ISSN 1042296X. doi: 10.1109/70.294210. 51, 53

[58] Yukihisa Karako, Shinji Kawakami, Keisuke Koyama, Makoto Shimojo, Taku Senoo, and Masatoshi Ishikawa. High-speed ring insertion by dynamic observable contact hand. In *IEEE International Conference on Robotics and Automation*. IEEE, 2019. 6, 7

[59] Branko Karan and Miomir Vukobratović. Calibration and accuracy of manipulation robot models—An overview. *Mechanism and Machine Theory*, 29(3):479–500, 1994. 59

[60] J.P. Karidis, G. McVicker, J.P. Pawletko, L.C. Zai, M. Goldowsky, R.E. Brown, and R.R. Comulada. The Hummingbird minipositioner-providing three-axis motion at 50 g's with low reactions. In *IEEE International Conference on Robotics and Automation*, pages 685–692, Nice, France, 1992. IEEE Comput. Soc. Press. ISBN 978-0-8186-2720-0. doi: 10.1109/ROBOT.1992.220288. 7

[61] Nathan Kau, Aaron Schultz, Natalie Ferrante, and Patrick Slade. Stanford doggo: An open-source, quasi-direct-drive quadruped. In *IEEE International Conference on Robotics and Automation*. IEEE, 2019. 6

[62] H. Kazerooni. Direct-drive active compliant end effector (active RCC). *IEEE Journal of Robotics and Automation*, 4(3):324–333, June 1988. ISSN 0882-4967. doi: 10.1109/56.793. 6, 30

[63] Gavin Kenneally, Avik De, and D. E. Koditschek. Design Principles for a Family of Direct-Drive Legged Robots. *IEEE Robotics and Automation Letters*, 1(2):900–907, July 2016. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2016.2528294. 2, 6, 12, 21, 22, 30, 31, 42, 50

[64] Gavin Kenneally, Wei-Hsi Chen, and Daniel E Koditschek. Actuator transparency and the energetic cost of proprioception. In *International Symposium on Experimental Robotics*, pages 485–495. Springer, 2018. 10, 16

[65] Oussama Khatib. Inertial Properties in Robotic Manipulation: An Object-Level Framework. *The International Journal of Robotics Research*, 14(1):19–36, February 1995. ISSN 0278-3649. doi: 10.1177/027836499501400103. 8, 15, 25

[66] Byeong-Sang Kim and Jae-Bok Song. Object grasping using a 1 DOF variable stiffness gripper actuated by a hybrid variable stiffness actuator. In *IEEE International*

Conference on Robotics and Automation, pages 4620–4625. IEEE, 2011. 7

[67] Robin Jeanne Kirschner, Nico Mansfeld, Saeed Abdolshah, and Sami Haddadin. Experimental Analysis of Impact Forces in Constrained Collisions According to ISO/TS 15066. In *IEEE International Conference on Intelligence and Safety for Robotics*, pages 1–5, Tokoname, Japan, March 2021. IEEE. ISBN 978-1-66543-862-9. doi: 10.1109/ISR50024.2021.9419494. 27

[68] Ghani KISSOUM and Véronique PERDEREAU. Simultaneous Tactile Localization And Reconstruction of an Object During Robotic Manipulation. In *International Conference on Advanced Robotics*, pages 948–954, December 2021. doi: 10.1109/ICAR53236.2021.9659354. 76

[69] G. S. Koonjul, G. J. Zeglin, and N. S. Pollard. Measuring contact points from displacements with a compliant, articulated robot hand. In *IEEE International Conference on Robotics and Automation*, pages 489–495, 2011. doi: 10.1109/ICRA.2011.5980463. 51

[70] Dragan Kostic, Bram De Jager, Maarten Steinbuch, and Ron Hensen. Modeling and identification for high-performance robot control: An RRR-robotic arm case study. *IEEE Transactions on Control Systems Technology*, 12(6):904–919, 2004. 59

[71] Oliver Kroemer, Christian Daniel, Gerhard Neumann, Herke van Hoof, and Jan Peters. Towards learning hierarchical skills for multi-phase manipulation tasks. In *IEEE International Conference on Robotics and Automation*, pages 1503–1510, May 2015. doi: 10.1109/ICRA.2015.7139389. 74

[72] Przemyslaw A. Lasota, Gregory F. Rossano, and Julie A. Shah. Toward safe close-proximity human-robot interaction with standard industrial robots. In *IEEE International Conference on Automation Science and Engineering*, pages 339–344, August 2014. doi: 10.1109/CoASE.2014.6899348. 9

[73] D. A. Lawrence. Stability and transparency in bilateral teleoperation. *IEEE Transactions on Robotics and Automation*, 9(5):624–637, October 1993. ISSN 1042-296X. doi: 10.1109/70.258054. 8, 10

[74] Xiong Li, Wenjie Chen, Wei Lin, and Kin Huat Low. A variable stiffness robotic gripper based on structure-controlled principle. *IEEE Transactions on Automation Science and Engineering*, 15(3):1104–1113, 2017. 7

[75] Michael A. Lin, Rachel Thomasson, Gabriela Uribe, Hojung Choi, and Mark R. Cutkosky. Exploratory Hand: Leveraging Safe Contact to Facilitate Manipulation in Cluttered Spaces. *IEEE Robotics and Automation Letters*, 6(3):5159–5166, July 2021. ISSN 2377-3766. doi: 10.1109/LRA.2021.3068941. 10

[76] R. D. Lorenz, J. J. Zik, and D. J. Sykora. A direct-drive, robot parts, and tooling gripper with high-performance force feedback control. *IEEE Transactions on Industry Applications*, 27(2):275–281, March 1991. ISSN 0093-9994. doi: 10.1109/28.73611. 6

[77] T. Lozano-Perez, J. Jones, E. Mazer, P. O'Donnell, W. Grimson, P. Tournassoud, and A. Lanusse. Handey: A robot system that recognizes, plans, and manipulates. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 843–849, March 1987. doi: 10.1109/ROBOT.1987.1087847. 65

[78] Kevin M. Lynch and Matthew T. Mason. Dynamic Underactuated Nonprehensile Manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*,

pages 889–896, 1996. 39

[79] Kevin M. Lynch and Matthew T. Mason. Stable pushing: Mechanics, controllability, and planning. 15(6):533–556, December 1996. 65

[80] R. R. Ma, L. U. Odhner, and A. M. Dollar. A modular, open-source 3D printed underactuated hand. In *IEEE International Conference on Robotics and Automation*, pages 2737–2743, May 2013. doi: 10.1109/ICRA.2013.6630954. 6

[81] Lucas Manuelli and Russ Tedrake. Localizing external contact using proprioceptive sensors: The Contact Particle Filter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5062–5069, Daejeon, South Korea, October 2016. IEEE. ISBN 978-1-5090-3762-9. doi: 10.1109/IROS.2016.7759743. 51

[82] Matthew T. Mason. Compliance and Force Control for Computer-Controlled Manipulators. *IEEE Trans on Systems, Man, and Cybernetics*, 11(6):418–432, 1981. 5, 9, 77

[83] Matthew T. Mason. *Manipulator Grasping and Pushing Operations*. PhD thesis, 1982. 65

[84] Matthew T. Mason and Kevin M. Lynch. Dynamic Manipulation. In *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 152–159, 1993. 76

[85] Matthew T. Mason, Siddhartha S. Srinivasa, Andres Vazquez, and Alberto Rodriguez. Generality and simple hands. in review, 2010. 65

[86] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *IEEE International Conference on Robotics and Automation*, pages 2520–2525. IEEE, 2011. 76

[87] Amirhossein H Memar, Nicholas Mastronarde, and Ehsan T Esfahani. Design of a novel variable stiffness gripper using permanent magnets. In *IEEE International Conference on Robotics and Automation*, pages 2818–2823. IEEE, 2017. 7

[88] C. Mittermayer and A. Steininger. On the determination of dynamic errors for rise time measurement with an oscilloscope. *IEEE Transactions on Instrumentation and Measurement*, 48(6):1103–1107, December 1999. ISSN 0018-9456. doi: 10.1109/19.816121. 33

[89] Gareth J Monkman, Stefan Hesse, Ralf Steinmann, and Henrik Schunk. *Robot Grippers*. John Wiley & Sons, 2007. 6

[90] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, first edition, December 2017. ISBN 978-1-315-13637-0. doi: 10.1201/9781315136370. 51, 52, 56

[91] A. Namiki, Y. Imai, M. Ishikawa, and M. Kaneko. Development of a high-speed multifingered hand system and its application to catching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2666–2671, Las Vegas, Nevada, USA, 2003. IEEE. ISBN 978-0-7803-7860-5. doi: 10.1109/IROS.2003.1249273. 6, 7

[92] Yoshiyuki Ohmura and Yasuo Kuniyoshi. Humanoid robot which can lift a 30kg box by whole body contact and tactile feedback. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1136–1141, 2007. 51

[93] A.M. Okamura, N. Smaby, and M.R. Cutkosky. An overview of dexterous manipula-

tion. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 255–262 vol.1, April 2000. doi: 10.1109/ROBOT.2000.844067. 65

[94] Peter Pastor, Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, and Stefan Schaal. Skill learning and task outcome prediction for manipulation. In *IEEE International Conference on Robotics and Automation*, pages 3828–3834, Shanghai, China, May 2011. IEEE. ISBN 978-1-61284-386-5. doi: 10.1109/ICRA.2011.5980200. 66

[95] Radhen Patel and Nikolaus Correll. Integrated force and distance sensing using elastomer-embedded commodity proximity sensors. In *Proceedings of Robotics: Science and Systems*. Robotics: Science and Systems Foundation, 2016. ISBN 978-0-9923747-2-3. doi: 10.15607/RSS.2016.XII.035. 39

[96] Richard Paul. Modelling, Trajectory Calculation and Servoing of a Computer Controlled Arm:. Technical report, Defense Technical Information Center, Fort Belvoir, VA, November 1972. 39, 65

[97] C. Piazza, G. Grioli, M.G. Catalano, and A. Bicchi. A Century of Robotic Hands. *Annu. Rev. Control Robot. Auton. Syst.*, 2(1):1–32, May 2019. ISSN 2573-5144, 2573-5144. doi: 10.1146/annurev-control-060117-105003. 6, 7

[98] Mihail Pivtoraiko and Alonzo Kelly. Kinodynamic motion planning with state lattice motion primitives. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2172–2179, September 2011. doi: 10.1109/IROS.2011.6094900. 66

[99] G.A. Pratt and M.M. Williamson. Series elastic actuators. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 399–406, 1995. ISBN 978-0-8186-7108-1. doi: 10.1109/IROS.1995.525827. 5, 20

[100] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: An open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, volume 3, page 5. Kobe, 2009. 35

[101] M.H. Raibert and J.J. Craig. Hybrid Position/Force Control of Manipulators. *Transactions of the ASME; Journal of Dynamic Systems, Measurement and Control*, 102, June 1981. 5

[102] A. Rao, D. Kriegman, and K.Y. Goldberg. Complete algorithms for feeding polyhedral parts using pivot grasps. 12(2):331–342, April 1996. ISSN 1042-296X. 65

[103] K. Renu, N. Krishna Kumari, and D.S.G. Krishna. Sensorless Control of Permanent Magnet Synchronous Motor with Flux Weakening Operation for Washing Machine Application. In *2018 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, pages 1–6, December 2018. doi: 10.1109/PEDES.2018.8707558. 78

[104] Robotiq. 2F-85 and 2F-140 grippers. 2019. URL https://robotiq.com/products/2f85-140-adaptive-robot-gripper. 46

[105] David Rollinson, Yigit Bilgen, Ben Brown, Florian Enner, Steven Ford, Curtis Layton, Justine Rembisz, Mike Schwerin, Andrew Willig, Pras Velagapudi, and Howie Choset. Design and architecture of a series elastic snake robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4630–4636, Chicago, IL, USA, September 2014. IEEE. ISBN 978-1-4799-6934-0 978-1-4799-6931-9. doi: 10.1109/IROS.2014.6943219. 37

[106] Elias Saerens, Stein Crispel, Pablo López García, Tom Verstraten, Vincent Ducas-

tel, Bram Vanderborght, and Dirk Lefeber. Scaling laws for robotic transmissions. *Mechanism and Machine Theory*, 140:601–621, October 2019. ISSN 0094114X. doi: 10.1016/j.mechmachtheory.2019.06.027. 6, 21

[107] Elias Saerens, Stein Crispel, Pablo Lopez Garcia, Vincent Ducastel, Jarl Beckers, Joris De Winter, Raphaël Guy Furnémont, Bram Vanderborght, Tom Verstraten, and Dirk Lefeber. Scaling laws for Parallel Motor-Gearbox arrangements. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6339–6346, 2020. 6, 21, 22

[108] S Salcudean and Chae An. On the control of redundant coarse-fine manipulators. In *IEEE International Conference on Robotics and Automation*, pages 1834–1840. IEEE, 1989. 77

[109] S. Salcudean and R. L. Hollis. A magnetically levitated fine motion wrist: Kinematics, dynamics and control. In *IEEE International Conference on Robotics and Automation*, pages 261–266 vol.1, April 1988. doi: 10.1109/ROBOT.1988.12058. 6

[110] Septimiu E Salcudean, Ming Zhu, Wen-Hong Zhu, and Keyvan Hashtrudi-Zaad. Transparent bilateral teleoperation under position and rate control. *The International Journal of Robotics Research*, 19(12):1185–1202, 2000. 8

[111] J Kenneth Salisbury Jr. *Kinematic and Force Analysis of Articulated Hands*. PhD thesis, Stanford University, 1982. 37, 65

[112] H. Sato, Y. Yamakawa, T. Senoo, and M. Ishikawa. Development of a high-speed, high-accuracy robot hand for micromanipulation. In *IEEE International Conference on Robotics and Biomimetics*, pages 1535–1541, December 2016. doi: 10.1109/ROBIO.2016.7866545. 7

[113] SCHUNK GmbH and Co. KG. WSG gripper. January 2019. URL https://schunk.com/us_en/gripping-systems/series/wsg/. 7, 33, 46

[114] Sangok Seok, Albert Wang, David Otten, and Sangbae Kim. Actuator design for high force proprioceptive control in fast legged locomotion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1970–1975, Vilamoura-Algarve, Portugal, October 2012. ISBN 978-1-4673-1736-8 978-1-4673-1737-5 978-1-4673-1735-1. doi: 10.1109/IROS.2012.6386252. 6, 10, 12, 21, 22

[115] Sangok Seok, Albert Wang, Meng Yee Michael Chuah, Dong Jin Hyun, Jongwoo Lee, David M Otten, Jeffrey H Lang, and Sangbae Kim. Design principles for energy-efficient legged locomotion and implementation on the MIT Cheetah robot. *IEEE/ASME Transactions on Mechatronics*, 20(3):1117–1129, 2015. 2, 6, 12, 21

[116] Michael Shomin and Ralph Hollis. Differentially flat trajectory generation for a dynamically stable mobile robot. In *IEEE International Conference on Robotics and Automation*, pages 4467–4472. IEEE, 2013. 76

[117] Jean-Pierre Sleiman, Jan Carius, Ruben Grandia, Martin Wermelinger, and Marco Hutter. Contact-Implicit Trajectory Optimization for Dynamic Object Manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6814–6821, November 2019. doi: 10.1109/IROS40897.2019.8968194. 74

[118] Barry Spletzer. Scaling laws for mesoscale and microscale systems. Technical Report SAND99-2172J, Sandia National Labs., Albuquerque, NM (US), 1999. 6, 12, 21

[119] Michael Strohmayr. Artificial skin in robotics. *Doktorarbeit, Karlsruhe Institute of*

Technology (KIT), 2012. 51

[120] T-Motor. T-Motor GB54-2, October 2017. URL http://store-en.tmotor.com/goods.php?id=445. 30

[121] TexasInstruments. InstaSPIN BLDC motor control solutions. October 2017. URL http://www.ti.com/instaspin. 32

[122] P. Tournassoud, T. Lozano-Perez, and E. Mazer. Regrasping. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 1924–1928, March 1987. doi: 10.1109/ROBOT.1987.1087910. 65

[123] W.T. Townsend and J.K. Salisbury. Mechanical bandwidth as a guideline to high-performance manipulator design. In *IEEE International Conference on Robotics and Automation*, pages 1390–1395, Scottsdale, AZ, USA, 1989. IEEE Comput. Soc. Press. ISBN 978-0-8186-1938-0. doi: 10.1109/ROBOT.1989.100173. 6

[124] Nathan T Ulrich. *Grasping with Mechanical Intelligence*. PhD thesis, University of Pennsylvania, Philadelphia, PA, 1989. 5

[125] J. Urata, T. Hirose, Y. Namiki, Y. Nakanishi, I. Mizuuchi, and M. Inaba. Thermal control of electrical motors for high-power humanoid robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2047–2052, September 2008. doi: 10.1109/IROS.2008.4651110. 37

[126] Junichi Urata and Yuto Nakanishi. Water-cooled motor structure and water-cooled housing, September 2017. URL https://patents.google.com/patent/US9768662B2/en. 37

[127] Jonathan Vorndamme, Moritz Schappler, and Sami Haddadin. Collision detection, isolation and identification for humanoids. In *IEEE International Conference on Robotics and Automation*, pages 4754–4761, 2017. 51

[128] I.D. Walker. Impact configurations and measures for kinematically redundant and multiple armed robot systems. *IEEE Transactions on Robotics and Automation*, 10 (5):670–683, October 1994. ISSN 2374-958X. doi: 10.1109/70.326571. 8, 15

[129] Sean Wang, Ankit Bhatia, Matthew T Mason, and Aaron M Johnson. Contact localization using velocity constraints. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020. 4, 10, 51, 55, 62, 76

[130] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim. Proprioceptive Actuator Design in the MIT Cheetah: Impact Mitigation and High-Bandwidth Physical Interaction for Dynamic Legged Robots. *IEEE Transactions on Robotics*, 33(3): 509–522, June 2017. ISSN 1941-0468. doi: 10.1109/TRO.2016.2640183. 6, 8, 15

[131] D. E. Whitney. Quasi-static assembly of compliantly supported rigid parts. *ASME Journal of Dynamic Systems, Measurement, and Control*, 104:65–77, March 1983. 5

[132] D.E. Whitney. Force feedback control of manipulator fine motions. *Journal of Dynamic Systems, Measurement, and Control*, 99:91, 1977. 5

[133] P.M. Will and D.D. Grossman. An Experimental System for Computer Controlled Mechanical Assembly. *IEEE Transactions on Computers*, C-24(9):879–888, September 1975. ISSN 1557-9956. doi: 10.1109/T-C.1975.224333. 9

[134] Tomoaki Yoshikai, Hiroko Fukushima, Marika Hayashi, and Masayuki Inaba. Development of soft stretchable knit sensor for humanoids' whole-body tactile sensibility.

In *IEEE/RAS International Conference on Humanoid Robots*, pages 624–631, 2009. 51

[135] Tsuneo Yoshikawa. Manipulability of Robotic Mechanisms. *The International Journal of Robotics Research*, 4(2):3–9, June 1985. ISSN 0278-3649. doi: 10.1177/027836498500400201. 15

[136] Jiaji Zhou, Robert Paolini, Aaron M Johnson, J Andrew Bagnell, and Matthew T Mason. A Probabilistic Planning Framework for Planar Grasping Under Uncertainty. *IEEE Robotics and Automation Letters*, 2(4):2111–2118, 2017. 74

[137] Wen-Hong Zhu and Tom Lamarche. Velocity estimation by using position and acceleration sensors. *IEEE Transactions on Industrial Electronics*, 54(5):2706–2715, 2007. 62

[138] Michael Zinn, Bernard Roth, Oussama Khatib, and J. Kenneth Salisbury. A New Actuation Approach for Human Friendly Robot Design. *The International Journal of Robotics Research*, 23(4-5):379–398, April 2004. ISSN 0278-3649, 1741-3176. doi: 10.1177/0278364904042193. 5