#### Yale University

## EliScholar – A Digital Platform for Scholarly Publishing at Yale

Yale Graduate School of Arts and Sciences Dissertations

Spring 2022

# Whole-Hand Robotic Manipulation with Rolling, Sliding, and Caging

Walter Gottlieb Bircher Yale University Graduate School of Arts and Sciences, waltergbircher@gmail.com

Follow this and additional works at: https://elischolar.library.yale.edu/gsas\_dissertations

#### **Recommended Citation**

Bircher, Walter Gottlieb, "Whole-Hand Robotic Manipulation with Rolling, Sliding, and Caging" (2022). *Yale Graduate School of Arts and Sciences Dissertations*. 560. https://elischolar.library.yale.edu/gsas\_dissertations/560

This Dissertation is brought to you for free and open access by EliScholar – A Digital Platform for Scholarly Publishing at Yale. It has been accepted for inclusion in Yale Graduate School of Arts and Sciences Dissertations by an authorized administrator of EliScholar – A Digital Platform for Scholarly Publishing at Yale. For more information, please contact elischolar@yale.edu.

#### Abstract

#### Whole-Hand Robotic Manipulation with Rolling, Sliding, and Caging

#### Walter Gottlieb Bircher

#### 2022

Traditional manipulation planning and modeling relies on strong assumptions about contact. Specifically, it is common to assume that contacts are fixed and do not slide. This assumption ensures that objects are stably grasped during every step of the manipulation, to avoid ejection. However, this assumption limits achievable manipulation to the feasible motion of the closed-loop kinematic chains formed by the object and fingers. To improve manipulation capability, it has been shown that relaxing contact constraints and allowing sliding can enhance dexterity. But in order to safely manipulate with shifting contacts, other safeguards must be used to protect against ejection. "Caging manipulation," in which the object is geometrically trapped by the fingers, can be employed to guarantee that an object never leaves the hand, regardless of constantly changing contact conditions. Mechanical compliance and underactuated joint coupling, or carefully chosen design parameters, can be used to passively create a caging grasp – protecting against accidental ejection – while simultaneously manipulating with all parts of the hand. And with passive ejection avoidance, hand control schemes can be made very simple, while still accomplishing manipulation. In place of complex control, better design can be used to improve manipulation capability-by making smart choices about parameters such as phalanx length, joint stiffness, joint coupling schemes, finger frictional properties, and actuator mode of operation. I will present an approach for modeling fully actuated and underactuated whole-hand-manipulation with shifting contacts, show results demonstrating the relationship between design parameters and manipulation metrics, and show how this can produce highly dexterous manipulators.

## Whole-Hand Robotic Manipulation with Rolling, Sliding, and Caging

A Dissertation Presented to the Faculty of the Graduate School of Yale University

> in Candidacy for the Degree of Doctor of Philosophy

> > by

Walter Gottlieb Bircher Dissertation Director: Aaron Michael Dollar May, 2022 ©2022 by Walter Gottlieb Bircher

All rights reserved.

"This has been said so many times that I'm not sure if it matters." -Patrick Stump

1	INT	RODUCTION	. 1
	1.1	Motivation	. 1
	1.2	Objective	. 1
	1.3	Outline	. 2
2	CAG	GING MANIPULATION ENERGY MODEL	. 3
	2.1	Caging Manipulation Model	. 5
	2.1.1	Caging Condition	. 7
	2.1.2	2 Caged-Object-Contact Space	. 8
	2.2	Manipulable Cage	. 8
	2.3	Case Study	11
	2.4	Experimental Procedure	11
	2.5	Actuation Space Exploration	13
	2.6	Object Workspace Exploration	13
	2.7	Manipulation Workspaces	15
	2.8	Simulation Comparison	18
	2.9	Conclusion	20
3	Ene	ERGY MODEL CONTROL	21
	3.1	Energy Model	24
	3.2	Energy Gradient-Based Graph	28

	3.2.1	Graph Construction	. 28
	3.2.2	Motor Actuation, Planning, and Execution	. 29
	3.2.3	System Implementation	. 30
3.3	3 (	Graph Evaluation and Experiments	. 32
	3.3.1	Graph Evaluation	. 35
	3.3.2	Experiments	. 35
3.4	4 C	Conclusion	. 36
4	DESIG	GN FOR PASSIVE CAGING	. 37
4.	1 U	Inderactuated Hand Simulation	. 39
	4.1.1	Underactuated Hand Model	. 40
	4.1.2	Free-swing Trajectory Simulation	. 40
	4.1.3	Cage Quality	. 43
	4.1.4	Design Parameterization	. 44
4.2	2 E	Design Parameterization Results	. 46
	4.2.1	Effect of link lengths and palm width	. 46
	4.2.2	Effect of spring stiffness ratio and pulley ratio	. 47
	4.2.3	Best gripper for caging object acquisition	. 48
4.	3 (	Conclusion and Design Principles	. 48
5	Ener	GY MODEL DESIGN FOR PLANAR CAGING MANIPULATION	. 49
5.	1 E	Energy-based Model Formulation	. 49

5.1.1	Forward Motion Model	. 52
5.1.2	System Kinematics	. 53
5.1.3	Energy Minimization	. 56
5.1.4	Energy Fields	. 58
5.1.5	Gradient of Energy Map	. 58
5.1.6	Manipulation Metric	. 59
5.1.7	Caging Metric	. 60
5.1.8	Kinematic Topology Enumeration	. 61
5.1.9	Design Space Variation	. 62
5.1.10	Object Size, Shape, and Pose Variation	. 64
5.1.11	Simulated Actuation	. 65
5.2 R	esults	. 67
5.2.1	Design Space Search	. 67
5.2.2	The Model W Hand	. 68
5.2.3	Workspace Evaluation with Test Objects	. 70
5.2.4	Real World Manipulation Scenario	. 71
5.3 C	onclusion	. 74
5.3.1	Design Space Search	. 74
5.3.2	Experimentation	. 75
Ener	GY MODEL DESIGN FOR SPATIAL CAGING MANIPULATION	. 76

6

	6.1	Energy-based Forward Motion Model	. 79
	6.1.1	1 System Kinematics	80
	6.1.2	2 Energy Minimization	. 82
	6.1.3	3 Energy Fields	83
	6.1.4	4 Gradient of Energy Map	. 84
	6.1.5	5 Manipulation Metric	85
	6.2	Simulation of Robotic Hands	86
	6.3	Simulation Results	. 88
	6.4	The Model B	88
	6.5	Experimental Results	. 90
	6.6	Conclusion	.94
7	Con	NCLUSION	.97
	7.1	Summary	. 97
	7.2	Lessons Learned and Future Work	. 98

# LIST OF FIGURES

2.1.	Manipulated object (shown in red) can be caged via point obstacles (a) or
	hand/gripper components (b)
2.2.	The caging capability of a planar hand configuration can be efficiently
	approximated by the largest inscribed circle with diameter $L_{obj}$ and the smallest
	escape opening length $L_{open}$ of the grasp polygon: the region bounded by the hand
	components
2.3.	Object energy field for caging manipulation evaluated for a simple gripper utilizing
	a single two-link, underactuated finger and a static thumb)27
2.4.	Various object geometries evaluated by the proposed caging manipulation
	experimental study
2.5.	The T42b hand design used in this study. Aruco fiducial markers were attached to
	the object and finger joints, and a webcam was mounted above the test fixture to
	track the system behavior
2.6.	Summary of an evaluated motion trajectory: from an initial point in actuation space
	(a), the hand is commanded to the target point in actuation space (b), and after the
	motion concludes, an open-loop, torque-based squeezing operation is commanded
	to ensure contact
2.7.	Experimental results for the circle object geometry and the T42b
20	Experimental results for the egg object geometry and the T42h 21
2.0.	Experimental results for the egg object geometry and the 1420

3.2. Given the geometry of a hand and corresponding object, we compute offline a library of energy maps—one for each unique combination of actuation inputs, over 3.3. The three largest strongly connected components (cc) of the graph created for the 18mm object, shaded according to the number of edges connected to each node..48 3.4. The normalized summed total edge distance of all shortest paths possible from a 3.5. Energy gradients help define desired non-linear trajectories when point-to-point 3.6. Example trajectory moving the object from the left to right side of the workspace. (Top) The teal trajectory indicates the shortest path found to the goal position (green), along waypoints (pink). Throughout our progression from (a) - (d), the system determines that we have deviated off the desired path in (b), so an updated path is formulated in (c), and executed in (d) until our goal position is reached. (Bottom) Energy gradient maps are evaluated at each time step, selecting the gradient that is instantaneously closest to our desired object direction. All paths are outlined in red, whereas the green arrow indicates our desired direction and the blue 4.1. Design parameter selection determines the largest opening between the fingers at contact, the number of reachable object positions, and in turn the passive caging 

4.2. In this work, caging quality is defined as the length of the smallest opening between the fingers. This can either be the distance between the fingertips, the distance between a fingertip and opposing link, or zero if the fingers interdigitat...59 4.3. Results of the design parameterization shows a range of caging ability (lighter is better) while varying link lengths, palm width, joint stiffness, and pulley radius. Each point represents the performance of a single hand design over its entire reachable workspace, averaged over the full range of object sizes. The lighter the color, the more high quality caging grasps a hand can make, based on the metric in 4.4. Left: The slice of design space containing the best gripper; Center & Right: The 5.1. Manipulability is derived from the convex hull formed by gradient vectors of the system's potential energy scalar fields. (A) Each contour plot shows the potential energy for a fixed square object configuration and a symmetric RR hand, given a distinct set of actuation inputs. The dashed lines represent the commanded finger positions, given the actuation inputs for each joint (in radians) listed below each contour plot, and the solid lines represent the realized finger positions, were they to close around the object in its fixed position. The red vectors emanating from the center of the object represent the gradients of the potential energy fields at the object's center point. The yellow vectors are the red vector scaled by the strength of the basic cage created around the object by the fingertips, a value from 0 (no cage) to 1 (fingers interdigitate). (B) The manipulation metrics are calculated by joining the tails of all vectors, calculating the convex hull of their tips, and finding

the radius of the largest origin-centered ball contained within their hull. Concretely, this radius is proportional to the minimum wrench the fingers can apply to the object in any direction (in the object's configuration space  $x-y-\beta$ ). (C) The manipulability (without caging) is represented by the radius of the red ball centered at the origin, and (D) the caging manipulability by the radius of the yellow ball...73 Simulation design parameters, visualized. (A) Two example topologies PPR and R are shown here with labeled dimensions. The distal-most link length is d, the angle of prismatic joints is  $\phi$ , middle links are of length l, and the palm width is p (one half of the palm is considered to be part of each finger). (B) The ten simulated objects are shown here at the same scale as the hand and grid in (C). Five distinct sizes of square objects and circular objects were simulated. The square objects were simulated at six orientations each, to capture the hand's ability to reorient them. (C) The objects were simulated at all positions shown in the 24x12 grid (example RR Simulation results from the design space search. Hand topologies are represented by simple models, where revolute joints are cylindrical and prismatic joints are rectangular prisms. (A) The RR topology is shown alongside the manipulability (H from Equation 10) scale, which goes from 0 (cannot manipulate any object in all directions) to 1 (best manipulability for all objects). (B) The manipulability design space (from convex hull of gradient vectors) for the RR topology, where each shaded pixel represents a different hand design consisting of parameters p (palm width), d (distal link length), and  $\phi$  (fixed prismatic joint angle). (C) The caging manipulability design space (convex hull of modified gradient vectors). See the

5.2.

5.3.

6.3.	Top Panel: the hands simulated in this work are shown and sorted by the number
	of actuators. An 'F' in a hand's name indicates that it is the fully actuated version
	of an underactuated hand from Yale OpenHand. Bottom Panel: The same hands are
	now sorted by their performance based on the manipulation metric $M_{avg}$ described
	in section II.F
6.4.	A rendering of the Model B hand. It is comprised of four fingers-two pairs of
	identical opposing fingers. One set is prismatic, the other rotates about the axis of
	the palm. Each finger has a proximal and distal link. The distal links of each finger
	interdigitate with the opposing finger
6.5.	The Model B can manipulate a wooden cube against gravity. A) The hand performs
	a "yaw" motion with the cube; B) The hand performs a "roll" motion with the cube;
	C) The hand performs a pinch to power transition against gravity, drawing the cube
	up from the support surface into a power grasp
6.6.	The Model B can manipulate objects with gravity into the palm. A) The hand
	performs a "yaw" motion with the wooden cube; B) The hand performs a left-right
	shift with a wooden cube; C) The hand performs a "roll" motion with the knitted
	cube; D) The hand performs a power to pinch transition against gravity with a ball.
6.7.	The Model B reoriented four painted cubes from G-R-A-B to Y-A-L-E,
	transitioning the hand to a vertical configuration before performing any
	manipulation primitives
6.8.	The Model B continuously reoriented the knitted cube while the WAM moved the
	hand through space, constantly changing the hand's orientation with respect to
	gravity

6.9. The Model B performed manipulation tasks with gravity pointing away from the palm. A) The hand begins to reorient four painted wooden cubes from Y-A-L-E to G-R-A-B; B) The hand performs a sequence of pre-determined open-loop manipulation primitives on the first cube, transitioning its front-facing side from the letter Y to the letter G; C) The hand completes the task of reorienting all cubes; D) The hand performs a "yaw" motion to the cube while moving through space; E) The hand performs a "yaw" motion.

### ACKNOWLEDGEMENTS

I would like to thank my advisor Aaron Dollar for believing in me and bringing me along on this journey and for giving me the freedom to explore ideas, good and bad, for as long as I wanted. It has been a challenging adventure, and he has been willing to say what needed to be said, even when it's not what I wanted to hear. I would also like to thank my other committee members Madhusudhan Venkadesan and Rebecca Kramer-Bottiglio for supporting me during this journey, and for mentoring their own students—my colleagues and friends—who have helped me through in so many ways. Big thanks to Matei Ciocarlie (Columbia) for his role as external reader! I also want to thank the outstanding postdocs that helped me become the researcher I am today. Namely, I want to thank Nicolas Rojas and Ad Spiers (both at Imperial College), Berk Calli (WPI), and Kaiyu Hang (Rice). Hang especially helped me grow my confidence and savvy as a researcher. Miss ya bud! I would also like to thank Shane Farritor (UNL) for teaching me how to design things and for urging me to apply for the NSF GRFP, without which I almost certainly would have not made it to Yale.

Next, I need to acknowledge Jude Clancy for expertly wiring up my ventricles and giving me a battery powered heart, Lawrence Young for determining that I needed one, Michele Spencer-Manzon for sequencing my exome and figuring out why, and Irwin Oh for giving me an extra 40mm of Achilles tendon per leg. It has been an absolutely wild ride these last few years, and I'm proud to say that I will not only leave Yale with a PhD, but also with a lifelong medical mystery solved and resolved (free of charge, shout out to the endowment!).

Last, I want to thank friends and family. Ali Yawar and I both arrived in New Haven in August of 2015, and met a few days later. We qualified for candidacy a few days apart. And now we're defending a day apart. Our birthdays are two days apart. Ali, I gotta thank you for being my

research twin, for going with me to the Jitterbus five times a day, and for taking two hour-long lunch breaks. You have always been so helpful any time I have some big technical question buzzing around in my head. You have always been there to talk about personal issues too, to get Blessings II Go, or to take apart printers that we found on the street. You are a great friend, and I can't even imagine what my PhD experience would have been without you. You're gonna do some amazing work at Harvard, ya freak.

Andy Morgan helped me turn many half-baked last minute publication ideas into ink, cross the graduation finish line, and have fun along the way. Andy, I gotta thank you for all the time and effort you put toward my success, and for turning me around when I got into a bad spot, time and time again. Once you make it out alive (I have no doubt will), give me a call and I'll tell Jeff directly to reserve a spot for you with a *hefty* sign-on bonus in Seattle, and we can rock the world of manipulation once more, this time in a fulfillment center (bring the Thinking Scooter). Finally, a huge thanks to my parents and sister, who were always there to tell me to keep going the numerous times that I seriously considered quitting. This PhD hasn't been easy, and I could not have done it without ya'll reminding me what matters in life.

## **1** INTRODUCTION

#### 1.1 Motivation

The work was motivated by observing human manipulation, which is highly dexterous and utilizes all surfaces of the fingers and palm, without enforcing fixed contact. With the presented work, we make progress towards that kind of dexterity in two main ways. First, we treat the hand-object-system holistically and utilize the observation that the system can be viewed in terms of total summed actuation effort and overall system energy, with corresponding variations in energy based on object location and configuration. Instead of the traditional method of calculating and controlling individual joints and actuators to result in a desired overall system configuration, we drive grasped objects to desired configurations by varying the potential energy in the system, forcing objects to follow energy gradients to a new state. This is accomplished without having to precisely model contacts or wrenches within the system, and is robust to the uncertainties that typically make producing controlled sliding or rolling extremely difficult. Second, we perform these manipulative actions while ensuring that the fingers "cage" the object [1, 2, 3]. This physically prevents the object from being ejected during manipulative movements that are generally risky to perform, such as stick/slip transitions.

#### 1.2 *Objective*

This work aims to investigate robotic manipulation with relaxed contact constraints from a design standpoint. Specifically, I am interested in "whole-hand-manipulation," which is manipulation with all parts of the hand—not just the fingertips. Manipulation of this kind often has multiple shifting points of contact, which necessitates sliding in order for the object to move with so many constraints on its motion. What are good design principles for performing manipulation in this way? If we perform sliding manipulation in a way that protects against object ejection (via caging grasps), can we enhance dexterity by using parts of the hand which are normally not used for manipulation? Can we passively enforce caging grasps using reconfiguration due to underactuation? My work here will fall under two goals. First, I plan to study how design can improve whole-hand-manipulation capability—including a hand's ability to translate an object within its grasp, to reorient an object within its grasp, and to do these things in a predictable manner without worry about ejection. Next, I will focus on the physical implementation and demonstration of sliding manipulation with underactuated hands. I will focus on both of these areas first in the context of underactuated hands, because underactuation can be leveraged to passively wrap around an object preventing ejection, and also because it allows the hands to be very simple to construct and control, with very few actuators, making them very robust. Next, I will apply these concepts to fully actuated hands.

#### 1.3 Outline

Chapter 2 introduces the caging energy model—the main theory underpinning the work in this dissertation. This theory was originally conceived in a paper I co-authored with Raymond R. Ma, who graduated from the GRAB Lab in December of 2016. Sections 2 through section 2.7 are from this paper [4], which was published in 2017, though the rest of the work in this dissertation is entirely my own work. This dissertation dovetails very nicely with the last chapter of Ray's, and can be viewed of a continuation of his work. The caging energy model essentially lays out a new way to think about manipulation, wherein caging (rather than force or form closure) guards against object ejection, while a potential energy view of the hand-object system is used to formulate a forward motion model that enables all surfaces of the fingers to be used for manipulation, rather than only the fingertips.

Chapter 3 is based on [5] and describes how the caging energy model can be used for closed loop control with visual servoing, translating objects within the caged grasp of an underactuated hand, the T42.

Chapter 4 is based on [6] and shows how the passive caging ability of underactuated hands can be improved through slight modifications of their design parameters.

Chapter 5 is based on [7] and reformulates the energy model using an optimization program, which dramatically speeds up the energy calculation. It also generalizes it to all planar serial-link fingers comprised of revolute and prismatic joints, both underactuated and fully actuated in nature. This reformulation enables a design space search resulting in a hand designed specifically for planar dexterous caging manipulation, the Model W.

Chapter 6 extends the formulation from chapter 5 to three-dimensions, and presents a highly dexterous hand for spatial caging manipulation, the Model B.

## 2 CAGING MANIPULATION ENERGY MODEL

Traditionally, within-hand manipulation has been modeled as a set of independently controlled fingers applying some controlled load, or wrench, to the object through a point contact. The fingers, usually represented as fully-actuated serial chains, are coordinated such that the desired stability and closure conditions are maintained as the system modulates the object pose [8], [9]. Despite extensive study in this area, physical implementations of dexterity with robotic hands has remained a major challenge, even with recent advances in hardware [10] and control fidelity [11], due to the high degree of complexity that this approach requires.

However, in-hand dexterity is not necessarily restricted to manipulation with fingertips exclusively [12], nor do hands need to always maintain closure conditions throughout the commanded task [13]. Consider the task of picking a screwdriver out of a cluttered bin and reorienting it properly into a secure grip for use. As the screwdriver is transitioned into the desired grasp, the number and nature of the contacts may change frequently, and there are likely many transitionary instances where even a small external wrench would be enough to eject the tool from the hand. It can be argued that the task is considered successful as long as the screwdriver can eventually be secured into the desired grasp.

We propose that ensuring repeatable object motion to an adequate range of poses for a set of system inputs can be a sufficient form of in-hand dexterity. The manner in which the object reaches the target pose or how the contact conditions change, disengage, and re-establish is not critical. By relaxing some of the constraints and requirements in traditional manipulation, other useful and simpler control strategies can be explored. Minimalist hands with simple control schemes can produce in-hand object motion.

Caging has been proposed as a robust method to bound the permissible range of poses for an object, albeit predominantly in the context of mobile, distributed robotics in the plane [14]. This approach simplifies the control scheme by permitting a limited amount of object free motion, recognizing that the system does not need to fully constrain the object during every phase of the task. Researchers have recently begun to acknowledge the potential utility for robotic hands performing caging primitives [15]–[18], but to our knowledge, none have evaluated this strategy for in-hand manipulation with physical robotic grippers.

In this paper, we detail the concept of whole-hand manipulation via caging, or *caging manipulation*, a dexterous control strategy especially well-suited for simple and/or underactuated

## TABLE 2.1 Object Energy Field Methodology Pseudocode

For each hand configuration  $q_{hand}$ : If  $isCaging(q_{hand})$ ,  $S_{caging} \leftarrow q_{hand}$ For each caging hand configuration  $q_{caging} \in S_{caging}$ For each actuated component  $AC_j$ : Calculate object contact subspace  $CS_j(q_{caging})$ Add  $\bigcap_j^p CS_j(q_{caging}), q_{caging}$  to object contact space OCFor each  $q_{obj}, q_{caging} \in OC$ : Calculate  $E_A(q_{caging})$ Add  $q_{obj}, \min(E_A(q_{caging}), EO(q_{obj}))$  to object energy field EO

grippers. Simple hands with a limited number of actuators can be configured such that even in the absence of force or form closure conditions, the object configuration space can be adequately limited and localized. This approach allows for robust open-loop control and can be shown to produce highly repeatable results without needing to characterize or track the contact conditions. Passive grasp adaptability through underactuated design can then be leveraged to further constrain the object after manipulation completes. This control methodology only requires the object to start within a local *capture region* [19] in the hand workspace, not any particular precision or power-grasp configurations, and it does not require any coordination between contact points. Extensive experimental results from the implementation of a planar underactuated hand and several object geometries will be presented to demonstrate the efficacy of this approach.

#### 2.1 Caging Manipulation Model

The manipulation strategy described in this paper combines previous work done on caging with linkage-based grippers [15], [16] and energy-based evaluation of underactuated hands' ability to hold and localize [20], [21]. An ideal, unconstrained object alone can be freely moved in its configuration space without any additional energy. Obstacles that are introduced into the object configuration space reduce its free workspace, and a hand can be thought of as a collection of rigid



Fig. 2.1. Manipulated object (shown in red) can be caged via point obstacles (a) or hand/gripper components (b)



Fig. 2.2. The caging capability of a planar hand configuration can be efficiently approximated by the largest inscribed circle with diameter  $L_{obj}$  and the smallest escape opening length  $L_{open}$ of the grasp polygon: the region bounded by the hand components

and compliant obstacles relative to the object that restricts its free space. Table 2.1 summarizes a methodology used to determine a hand's caging manipulation capability

Unlike the traditional notion of caging, which assumes static obstacles that generate fixed, inaccessible regions in the object workspace, we recognize that actuated components, such as finger phalanges and gripper surfaces, have an associated energy state [20], [21] that changes depending on their interactions with the object and other components in the system. Evaluating the energy state of the full hand-object system is a useful way to analyze how the multiple actuated components affect the object pose stability and motion.

In the absence of an object or other components, an actuated component in a conservative system should resolve to the lowest possible energy state. For example, a finger actuated with a

constant torque input will attempt to close fully, until it contacts an obstacle that restricts its motion or reaches a physical hard-stop. A system with multiple actuated components is expected to reconfigure towards the overall energy-minimal configuration that satisfies the necessary kinematic limits.

With respect to caging, there may be multiple energy-minimal system configurations for a given set of actuation inputs. However, for any given set of inputs, we can identify a stable *attractor region* in the object workspace towards which the manipulator is actively driving the object.

#### 2.1.1 Caging Condition

In summary, an object is caged if it cannot be moved to a point at infinity without intersecting other components in its workspace. Fig. 2.1 compares the traditional caging problem, commonly utilizing point-based obstacles, and the corresponding caging problem for a planar, two-finger hand, which can be represented by a set of serial chains. The generalized caging problem has been previously detailed and thoroughly investigated by several researchers [15], [18], [22].

For the simplified planar case, which will be the focus of this paper, the caging problem can be simplified to two sub-problems regarding the *grasp polygon*: finding the minimum opening length  $L_{Open}$  and the diameter of the maximal inscribed circle $L_{Obj}$ , as detailed in Fig. 2.2. The grasp polygon is the planar polygon with edges formed by the finger phalanges, the palm surface, and the edge between the pair of distal fingertips. For the common case with two opposition fingers, the minimal opening can be found as either the magnitude of the vector between the distal fingertips or the vector between a distal fingertip and an opposing phalanx, perpendicular to that phalanx. The maximal inscribed circle is found by iterating through all unique triplets of grasp polygon edges and finding the circles tangent to all three edges with centers lying within the grasp polygon. The resulting set of configurations  $S_{caging}$  is an approximation of the hand's caging capability and greatly reduces the number of system configurations that need to be evaluated for each object geometry.

#### 2.1.2 Caged-Object-Contact Space

It is more computationally efficient to focus our investigation of the hand-object system to the *caged object-contact space*, the set of object-hand configurations where the hand cages the object and work is being done on the actuators due to object contact. Though the analysis presented here will also account for configurations where the object does not necessarily make contact with the hand components, physical manipulation of a grasped object requires contact with actuated components.

For each caging configuration  $q_{caging}$  in  $S_{caging}$ , the object contact subspace  $CS_j(q_{caging})$ for the object and each actuated component  $AC_j$  can be determined by finding the object poses such that the object and component boundaries are coincident. The object-contact-space for the entire mechanism is then the intersection of the contact spaces  $\bigcap_{j=1}^{P} CS_j$  calculated for the actuated components of interest. For the planar, underactuated hand designs investigated in this study, that space is comprised of configurations where the object makes contact with both fingers but not necessarily all finger links. For more complex hands, especially ones where not all actuated components necessarily need to contact the object during manipulation, the different permutations of actuated components would need to be considered.

#### 2.2 Manipulable Cage

Instead of considering only rigid, immovable caging configurations, as is used in the conventional definition of caging, we propose the concept of a *manipulable cage* - caging

configurations that can be reconfigured into other caging or non-caging configurations with some non-zero work. Relative to the commanded reference inputs, the work done by the actuators can be mapped to the corresponding object configuration satisfying the contact constraints of the manipulator [20]. Mahler et al. [23] have presented a similar concept called energy-bounded caging - configurations which effectively cage the object with the assistance of some external force, such as gravity. In both of these definitions, an escape path in the energy field can be computed for a caged object, and the energy expenditure necessary to free the object is calculated. In our approach, however, we propose utilizing an understanding of the energy profile of the possible object configurations in order to modulate the object motion within the hand.

For a given reference value  $a_k$ , either position  $p_A$  or rotation  $\theta_A$ , the energy for the *k*th actuated component is:

$$E_{Ak}(p_k) = -F_{Ak}(p_k - p_{Ak})$$

$$1)$$

$$E_{Ak}(\theta_k) = -\tau_{Ak}(\theta_k - \theta_{Ak})$$

$$= -f_{Ak}r_{Ak}(\theta_k - \theta_{Ak})$$

$$2)$$

for actuation force  $f_A$  or actuation torque  $\tau_A$  and corresponding transmission radius  $r_A$ . In the context of hands, we assume that actuated components (most commonly finger phalanxes) can only push, not pull, so configurations with negative energy values, indicating configurations where the actuator can achieve the reference input without the associated actuated components making contact with the object, are treated as zero-energy configurations. For simplicity, the effects of passive elements like return springs or flexural stiffness on the energy state are assumed to be negligible relative to the actuator energy and are disregarded.



Fig. 2.3. Object energy field for caging manipulation evaluated for a simple gripper utilizing a single two-link, underactuated finger and a static thumb

The full system energy for each configuration is then the summation of the energy for all actuators:

$$E_A(a_{hand}) = \sum_k^N max(E_{Ak}(a_k), 0)$$
(3)

Under the assumptions of this model, each actuator is at its lowest energy configuration if it reaches its target commanded reference. In a multi-component system, interactions and interferences between components can make it impossible for each actuator to achieve its commanded reference. The system is expected to reconfigure towards the lowest energy configuration, of all the possible system configurations permitted by the components' geometries and respective workspaces. Evaluating this for all hand-object configurations in turn produces an object energy field detailing the system reconfiguration behavior in response to the actuation inputs. In this way, each caging manipulation primitive can establish a particular region of attraction in the object workspace. These details can be numerically extracted from the object energy field to provide more insight regarding the grasp stiffness and the effective bias that the hand applies to the object pose [24].

#### 2.3 Case Study

Fig. 4 presents an example object energy field, evaluated for a simple gripper, composed of a two-link underactuated finger and a static opposition thumb (Fig 2.3a), manipulating a circle object. The system energy across the object-contact space were evaluated for in increasing set of actuation reference values, and the plotted energy values (Fig. 2.3b) were normalized with respect to the maximum recorded system energy value. The model shows that, as expected, closing the finger pulls the object inwards, towards a *y*-value around 0.4 (the lighter regions in the plot), for an object with radius 0.175. By increasing the actuation input from 0.6 to 0.8, the lower-energy regions become smaller, indicating that the object is held in a more secure grasp. For actuation input 0.4, the smaller overall shaded region indicates that there are portions of the object-contact space where the reference position value for the actuator was not sufficient for the finger to close enough to make contact with the object.

#### 2.4 Experimental Procedure

To evaluate this manipulation strategy experimentally, we explored the planar, cagingmanipulation workspace for various object geometries (Fig. 2.4). The egg geometry was determined by a 25mm diameter circle and a 45mm diameter circle with a variable offset between the two. All objects were printed and had attachment points for fiducial markers. This paper



Fig. 2.4. Various object geometries evaluated by the proposed caging manipulation experimental study



Fig. 2.5. The T42b hand design used in this study. Aruco fiducial markers were attached to the object and finger joints, and a webcam was mounted above the test fixture to track the system behavior

focuses on results for the T42b hand, a tendon-driven design with a two-link underactuated finger and an opposing one-link thumb. The two fingers are each independently driven by a Dynamixel MX-28 smart servo. The design, detailed further in Fig. 2.5, is a variation of an underactuated design taken from the Yale OpenHand Project library [25]. To minimize friction, both the finger and object surfaces were ABS, 3D-printed with layers oriented in the same direction.

A Logitech C920 webcam was mounted above the test hand, which was fixtured in place. Aruco fiducial markers [26] were affixed to the test object centroid and ends of each finger link, via the revolute joint center where possible. Using Python and OpenCV for image capture and fiducial tracking, this setup could record at approximately 15 frames per second. For each commanded actuation input, the marker positions were tracked and recorded continuously throughout the motion. The object space reference frame was established at the midpoint between the two proximal finger joints.

#### 2.5 Actuation Space Exploration

To generate the viable actuation space for each object, we took advantage of underactuated hands' mechanical adaptability. Each actuator's operating range was first discretized, and for each actuator, it was driven to each discretized value via position-control. The opposing actuator was then commanded to close via a constant torque, and the actuator encoder values were recorded after the object and hand elements were fully constrained. This exploration excluded cases where the hand ejected the object during grasp acquisition or where the hand configuration was visually identified as a non-caging. The object was reset to the middle of the hand workspace between each grasp acquisition test. This sparse sampling of points in the actuator space was then interpolated to produce the set of actuator inputs used in the workspace evaluation.

#### 2.6 Object Workspace Exploration

The exploration procedure iterated through and tested all possible initial and target combinations of actuator inputs from the sampled actuation space. In order to account for contact variability and/or hysteresis in the pulley transmission, the actuators are initially driven to their



Fig. 2.6. Summary of an evaluated motion trajectory: from an initial point in actuation space (a), the hand is commanded to the target point in actuation space (b), and after the motion concludes, an open-loop, torque-based squeezing operation is commanded to ensure contact



Fig. 2.7. Experimental results for the circle object geometry and the T42b



Fig. 2.8. Experimental results for the egg object geometry and the T42b

target values in position-mode, and then switched to torque-mode to maximize contact. Fig. 2.6 summarizes these steps in a typical tested manipulation execution.

The actuations inputs calculated in the previous section are sufficient to keep the object within the grasp acquisition range and avoiding ejection. As a result, each workspace exploration could be run continuously. A typical exploration of the full actuation space evaluates ~160 independent motion trajectories, lasting a total duration of 20 minutes. At least two full workspace exploration trials were completed for each unique hand-object combination.

#### 2.7 Manipulation Workspaces

Examples of the object workspaces achievable through the proposed caging manipulation are shown in Fig. 2.7-2.8. The black points designate the final object grasp poses, and the grey points corresponds to all object poses during the execution of each caging manipulation move. The results for all the evaluated hand-object combinations are detailed further in Table 2.2. The achievable object workspaces ranged from 29 to 46mm in the x-direction, 5 to 26mm in the y-direction, and up to 0.97 rad in total reorientation.

As has been previously proposed in past work [20], [27], [28], the manipulation capability is determined by a combination of the hand's geometric design parameters and the object shape. In particular, the overall *xy* workspace was most limited for the rectangular and square objects, which were often aligned against the hand palm or a finger link. The egg-shaped and circle geometrys' curved surfaces made them easier to reconfigure within a grasp and avoid line-contacts with the finger-links or palm, made evident by the increased *xy* and reorientation workspaces.

Figure 2.9 presents an example object trajectory for the 55x45mm egg object superimposed on the recorded hand-object workspace. This specific trajectory shows a substantial translation in the horizontal direction. To locate this trajectory within the dataset, we divided the recorded object



TABLE 2.2

Fig. 2.9. Example trajectory comparing the recorded fiducial data and the video. Trajectory in the video overlay were manually identified from frame grabs. Automatic fiducial detection failed when manipulation speed was too high, so recorded fiducial data lacks some of the trajectory details between the initial and final grasp poses.

motions into left and right movements, and sorted them into bins with respect to object displacement. Then, the trajectory with the largest horizontal translation was selected from the
most populated bin. The evaluated object workspaces can be sampled to determine highly repeatable object trajectories that may be useful when planning manipulation strategies. The media attachment also included many additional examples of caging manipulation, executed during experimental studies as well as scripted tasks performed on a robotic arm manipulator.

#### 45mm Circle Workspace



Fig. 2.10. 3x3 grid of simulated object energy fields for the T42b hand and a circular object. The energy values for each actuation input are normalized with respect to the maximum energy configuration in that particular input pair. Symbols represent experimental object start points (small symbols) and end points (large symbols). For example, the object moves from small circle to large circle after the hand is commanded with the corresponding actuation input.

## 2.8 Simulation Comparison

Fig 2.10 shows experimental results collected from the methodology described in Section IV.D in the form of object start points (smaller symbols) and object end points (larger symbols)

resulting from actuating the hand according to the actuation input pair, which are superimposed on each corresponding energy field. In each case, the hand-object system settled such that the object came to rest near the lowest energy position in the simulated energy field, regardless of start position. The spread of endpoints in the [0.4, 0.4] subplot is because dual finger contact is never established with this actuation input pair (no dashed lines).

Fig 2.11 shows experimentally collected object manipulation trajectories superimposed on corresponding simulated actuator energy maps. For a given actuation input, all object trajectories terminate in nearly the same position, and orientation in the case of the square object, regardless of starting position. This position is in a low energy region of the simulated actuator energy workspace. Additionally, object trajectories roughly follow the gradients of the energy maps, as predicted.



Fig. 2.11. Actuation input [0.4, 0.7] experimental trajectories from arbitrary selected start points. Left: trajectories for 45mm circle, superimposed over the corresponding energy map. Center: trajectories for 45mm square, superimposed over the corresponding energy map. Right: orientation trajectories for the 45mm square object corresponding to the trajectories shown in the center panel.



Fig. 2.12. Summary of the difference in actuator inputs for a direct vs a gaited motion



Fig. 2.13. Comparison between the workspaces evaluated through direct actuator move commands vs gaited actuator move commands

# 2.9 Conclusion

In this paper, we analyzed *caging manipulation*, a manipulation primitive, which could be also be described as in-hand fumbling or shuffling. The caging characteristic allows for open-loop trajectories that avoid object ejection or loss of grasp without detailed knowledge of the contact conditions. This manipulation primitive was evaluated on a physical test setup for a set of object geometries and a planar, underactuated hand design with two-actuators. The experimental results showed that simple, open-loop actuator commands were sufficient to robustly manipulate objects within the hand workspace. Examples of open-loop gaiting motions, made possible by caging, were also demonstrated as a means of extending the manipulation workspace and compensating for different coefficients of friction. This may run counter to past, traditional approaches to dexterous manipulation, which requires object stability and well-maintained contact conditions within the grasp at all instances of the executed task.

While the proposed caging manipulation primitive can be applied on any hand design, it is particularly useful in underactuated hands, which are typically designed to passively cage around the object, regardless of the particularities of its geometry. Caging manipulation extends the underactuated hand's passive adaptation and applies a bias to the object, constrained to its allowable workspace relative to the hand. We hope that the robustness demonstrated by the experimental examples will encourage researchers to consider other manipulation primitives that relax grasp constraints where possible to enable more functionality in other useful ways.

# **3** ENERGY MODEL CONTROL

Manipulating a grasped object within the hand is an important functionality for many practical tasks, especially for instances where the grasp type must be changed without releasing the object, such as changing from a fingertip grasp to a palmar grasp. Nearly all within-hand manipulation (WIHM) tasks involve changing the contact location on the hand or object, which will typically involve some amount of sliding. Rather than directly modeling the complex frictional properties and behaviors at contact in these scenarios, we instead seek to create a scenario in which the object is passively prevented from being ejected from the grasp (i.e. it is "caged" grasp [29] [4]), and the manipulation is actively guides the object in the desired directions by shaping the potential energy of the underactuated fingers. In this way, we can ensure that the object moves

towards the desired target without needing precise information about the contact forces and frictional properties.

In this work, we present a within-hand manipulation approach that leverages a simple energy model based on caging grasps made by underactuated hands. Instead of explicitly modeling the contacts and dynamics in manipulation, we can calculate a map to describe the energy states of different hand-object configurations under an actuation input. Since the system intrinsically steers towards low energy states, the object's movement is uniquely described by the gradient of the energy map if the corresponding actuation is applied. Such maps are pre-calculated for a range of actuation inputs to represent the system's energy profile. We discretize the workspace into a grid and construct an energy gradient-based graph by locally exploring the gradients of the stored energy profile. Given a goal configuration of a simple cylindrical object, a sequence of actuation inputs can be calculated to manipulate it towards the goal by exploiting the connectivity in the graph. The proposed approach is experimentally implemented on a Yale T42 hand. Our evaluation results show that parts of the graph are well connected, explaining our ability to successfully plan and execute trajectories within the gripper's workspace.

Traditional approaches to this type of problem rely strong assumptions about the nature of contact—namely being able to precisely model the contacts between the robot and object in order to enable effective control [9][30][31][32]. By relaxing the rigid constraints in a grasp, objects can be manipulated by rolling contacts on its surface based on kinematic trajectory optimization [29]. In an object-centric formulation, a virtual-frame can be derived to enable impedance control to implicitly regulate contact forces during manipulation [33]. Using tactile feedback from the fingertips, grasp stability can be estimated online to inform the system so that force adaptation and finger gaiting can be utilized to prevent the system from dropping the object [34], [35]. Although

these approaches can sometimes successfully reconfigure the object within hand, they are vulnerable to external disturbances and require great mechanical and computational complexity to maintain the grasp and often fail due to uncertainty or errors in the required sensing and control.

Rather than using high Degree of Freedom (DOF) hands for manipulation, extrinsic dexterity has enabled simple grippers to reconfigure an object with larger motions by exploring external contacts [36]. To understand how an object can be manipulated by external pushing, motion cones have been proposed to represent feasible actions applicable to an object [37]. Moreover, by analyzing the geometries of objects to model the feasible translations and rotations of contacts on an object surface, dexterous manipulation graphs have been proposed to plan a sequence of pushing actions in a dual-arm formulation [38]. Nevertheless, this class of approaches requires complicated geometrical analysis of both the object and the environment, and generates large motions of the arm.

Instead of using a force feedback-based grasp for manipulation, caging grasps can be used as a very robust means to guard against external disturbances [23]. Using topological representations, neck or fork structures in an object can be detected to enable caging by simple grippers [39]. Moreover, loop-grasping can also be synthesized by modeling a Writhe Matrix between the loops in an object and robot links [40]. Caging has also been used for "blind" (open loop) within hand manipulation using a fully defined model of the hand [41]

In our previous work, based on caging grasps and the passive reconfigurability of underactuated hands, we have developed an energy map to implicitly represent the mapping between the hand-object configuration and the actuation inputs [4]. This enables us to understand the energy status of the hand-object system, as well as how the object can move towards a lower energy configuration under certain actuation inputs. Based on those energy maps, in this work we

develop a graph representation to model the connectivity among the object positions in the hand's workspace. In brief, based on a set of energy graphs calculated from different actuation inputs, the graph is constructed by exploring the map's gradient directions, along which the object moves under different actuations. The graph is then used to plan a sequence of actuation inputs to reconfigure the object towards a given goal configuration, while attempting to maintain the object in a caging grasp.

#### 3.1 Energy Model

In this paper we utilize a planar energy based caging model first presented in [4] to translate symmetric cylindrical objects. In short, this work combines a linkage based caging model with a method that computes the total energy of the hand-object system in each kinematically feasible state. Whereas a traditional caging model assumes immovable rigid obstacles, we instead acknowledge that obstacles can be moved for a cost. This assumption is valid for two reasons. First, it is valid because we apply this model to an underactuated hand with compliant elements each with a number of kinematically admissible configurations, and each storing different amounts of energy. Second, we consider actuators to be backdrivable, treating them as linear springs around their commanded setpoints. In other words, if you do work to rotate the shaft of an actuator operating in position control mode, you can change its position. With these assumptions in mind, we use an extended caging formulation, beyond the more traditional purely kinematic analysis, by also considering the energy associated with movable obstacles.

By definition an object is caged if it cannot be moved to a point at infinity without first intersecting other objects in its workspace. In general, this corresponds to a point contained within a closed, isolated volume in configuration space. In this work we narrow our scope to the case of a planar object being caged by the links of a planar gripper. We adopt notation from the caging formulation described in [16] and consider caging configurations that minimize the object's configuration space. We do this by making the strong assumption that there are no dissipative forces in our system, and that a stable grasp on an object, representing a single actuation input combination, is associated with an energy minimum configuration. In other words, we assume that for a given object position, there is some combination of actuator inputs that minimizes the system's energy, somewhere in the feasible range of joint configurations that adhere to the physical contact constraints between the links and the object. This allows us to consider a manipulable caging grasp, meaning that the hand can be reconfigured into other non-caging configurations with non-zero work done on the object. To formulate the energy minimization, we follow previous work from [4], computing energy values specifically for caged configurations of our system:

- $p_k$ : reference position for joint or element k, either  $p_A$  for linear actuators or  $\theta_A$  for rotary actuators, p for joints
- $f_A$ : force from actuator A (for linear actuators)
- $\tau_A$ : torque from actuator A (for rotary actuators)

The actuation energy associated with a given reference value  $a_k$  can be expressed as the following for rotary actuators:

$$E_{Ak}(\theta_k) = -\tau_{Ak}(\theta_k - \theta_{Ak})$$
$$= -f_{Ak}r_{Ak}(\theta_k - \theta_{Ak})$$
(1)

Or, in the case of linear actuators:

$$E_{Ak}(p_k) = -f_{Ak}(p_k - p_{Ak})$$
 (2)

Then, the total energy associated with a configuration of the hand is written as the summation of the energy for all actuators in the system:

$$E_A(a_{hand}) = \sum_{k}^{N} max(E_{Ak}(a_k), 0)$$
(3)

where  $a_k$  is the position controlled actuation input. The *max* function is used to select only positive energy values (see [4]). The system energy, which depends only on the configuration of the hand, is computed using 3 at each caged object *xy*-position in front of the hand. As the object is virtually placed throughout the workspace, the hand's configuration adjusts to maintain contact. Thus, the hand's configuration would change if the object were forcefully placed in its path, doing energy against the actuators. This workspace of energy values forms a contour plot, similar to that



Fig. 3.1. A Yale OpenHand T42 gripper and associated energy map (contour plot). The grid of gradient vectors of the energy map (red arrows) show the possible motions that can be applied to an object at each location with a hand, for a specific actuation input [0.4, 0.4].

shown in Fig. 3.1. We refer to this bounded contour plot containing system energy values as an Energy Map  $M_i = f(E_A(x, y)), \in \mathbb{R}^2$ . A single energy map can be computed for every combination of actuation inputs, as described in [4], and as illustrated in Fig. 3.2. We extend our previous work by numerically computing the gradient vector field  $\gamma_i$  of a simulated energy map

$$\gamma_{i} = -\nabla_{x,y} M_{i} \tag{4}$$

For a given hand-object configuration and a given actuator input set,  $\gamma_i$  can be visualized as a vector field overlaid on the workspace of the hand, with all vectors flowing towards the lowest system energy. An example is shown in Fig. 3.1.



Fig. 3.2. Given the geometry of a hand and corresponding object, we compute offline a library of energy maps—one for each unique combination of actuation inputs, over the range of possible actuation inputs for the motors of the hand.

#### 3.2 Energy Gradient-Based Graph

In this section, based on the set of obtained energy maps  $\mathcal{M} = \{M_1, ..., M_N\}$ , we will first introduce the construction of the energy gradient-based graph, and then use the constructed graph to plan actuation input sequences to move the object between positions in the workspace. In our case we utilized a set of n = 100 energy maps corresponding to 100 actuation input combinations.

#### 3.2.1 Graph Construction

In this section, based on the set of obtained energy maps  $\mathcal{M} = \{M_1, ..., M_N\}$ , we will first introduce the construction of the energy gradient-based graph, and then use the constructed graph to plan actuation input sequences to move the object between positions in the workspace. In our case we utilized a set of n = 100 energy maps corresponding to 100 actuation input combinations.

#### 3.2.2 Motor Actuation, Planning, and Execution

Having constructed the energy gradient-based graph *G*, we are now able to plan a sequence of actuation inputs in order to manipulate the object to translate along the edges in *E*. Concretely, letting  $n_s, n_g \in V$  be the initial and goal positions of the object, we aim to find a path  $\Pi =$  $\{n_s, \pi_1, ..., \pi_T, n_g\}, \pi_1 \in V$ , such that the goal position can be reached by executing the actuation input associated with each edge  $(\pi_t, \pi_{t+1})$ .

In reality, however, the start and goal positions  $n_s$ ,  $n_g$  are in continuous space and unlikely to be exactly on the grid. Therefore, before finding the path  $\Pi$  we snap the continuous positions to the grid points using the K-Nearest Neighbor (KNN) algorithm. In order to increase the planning success rate, rather than finding one nearest neighbor for each,  $n_s$  and  $n_g$  are snapped to m nearest neighbors by the function  $KNN(n_i, m)$  to generate a set of m start candidates  $\{n_s^1, \dots, n_s^m\}$ , and mgoal candidates  $\{n_g^1, \dots, n_g^m\}$  in the graph. The candidates are ordered by their distances to the original point. During planning, we try to find a path by iterating through each pair of start and goal candidates and returning a path as soon as the first path is found for a pair  $n_s^i, n_g^j$ .

For executing the path, in order to compensate for the inaccuracies caused by snapping and discretization, the motor actuation for the first edge  $(n_s^i, \pi_i)$  in  $\Pi$  is not directly found by its associated actuation input. Instead, we calculate the actuation input based on the original  $n_s$  and the first waypoint  $\pi_1$ . The index of the actuation input at the start point  $n_s$  is found by:

$$L^* = \underset{l}{\operatorname{argmin}} \delta(\overline{n_s \pi_1}, \gamma(M_l, n_s))$$
(6)

Furthermore, to compensate for noise and execution errors, although we have an entire path planned, we in practice execute only the first actuation input  $L^*$ . Thereafter, our system will re-

observe a new  $n_s$  and replan a path, from which again only the first actuation is executed. In this manner, our system will iteratively move the object towards the goal position, while being able to adjust its behavior on the way by online re-planning. The planning and execution based on the energy gradient-based graph is summarized in Algorithm 1.

Algorithm 1 Planning and Execution				
<b>Input:</b> $G = (E, V), n_g, m, maxIter$				
Output: Status				
1: while $maxIter \ge 0$ do				
2: $n_s \leftarrow observePosition()$				
3: <b>if</b> $  n_s - n_g   \leq \eta$ then				
4: return Success	▷ Reached			
5: end if				
6: $N_s \leftarrow KNN(n_s, m)$				
7: $N_g \leftarrow KNN(n_g, m)$	▷ Snapping			
8: $P = constructPairs(N_s, N_g)$				
9: for all $p \in P$ do				
10: $(n_s^i, n_g^j) \leftarrow p$				
11: $\pi = G.FindPath(n_s^i, n_g^j)$				
12: <b>if</b> $\pi$ . <i>found</i> () <b>then</b>				
13: $L^* \leftarrow GetActuation(n_s, \pi)$	)			
14: $Execute(L^*)$	▷ Execution			
15: <b>else</b>				
16: <b>return</b> Failure	⊳ No Path			
17: <b>end if</b>				
18: <b>end for</b>				
19: $maxIter = maxIter - 1$				
20: end while				
21: return Failure	⊳ Too Many Steps			

For path finding in the graph, we use breadth-first search to find the shortest path. We can see that the manipulation is limited to take at maximum *maxIter* steps. Once the difference between real-time observed object position and goal position is smaller than  $\eta$ , we consider it as a success. A failure can occur if a path cannot be found to connect start and goal, or the maximum number of executions has been exceeded.

#### 3.2.3 System Implementation

This framework was physically implemented and tested by utilizing an overhead camera observing the manipulation. First, state detection processes determined the physical location of the

gripper through the use of ArUco markers attached to the base frame of the hand. The origin of the workspace was then calculated, which is specified to be directly between the base frame joints of the hand. It is important to note that the object can move in the positive and negative x direction, but can only in the positive y. At this point, the object's physical location was then extracted with respect to the defined origin of the gripper. From this information, we were able to define the current location of the object which is utilized for planning a desired trajectory.

Three 3D printed cylindrical objects were manipulated and the system was evaluated. Radii corresponding to these objects were 18mm, 22mm, and 27mm, respectively. The gripper was attached to a physical support structure so that the manipulation workspace was parallel to the object's support plane. System state data was then evaluated online, observing object location, goal location, and corresponding nodes of interest in the graph. The system executed randomly selected goal locations, corresponding to random nodes in the graph. Since not all points are reachable, a validity check first determined if a path existed. If a path did not exist, a new random



Fig. 3.3. The three largest strongly connected components (cc) of the graph created for the 18mm object, shaded according to the number of edges connected to each node.

location in the workspace was selected. Otherwise, we executed the desired path determined by the graph.

## 3.3 Graph Evaluation and Experiments

In this section, we evaluate properties of our constructed graph to understand connectivity between different areas of the workspace. We follow this discussion with experiments conducted on a physical robotic hand, and evaluate successful trajectories found in our implementation.

To better understand the manipulation capability of specific hand-object systems, we used standard graph theory measures to evaluate the graphs constructed from the set of energy maps as described in Section II. First, we considered the connectivity of the graph, to learn where the object might be more freely manipulated within the hand. Specifically, we found the strongly connected



Fig. 3.4. The normalized summed total edge distance of all shortest paths possible from a given start node for the smallest object (18mm).



Fig. 3.5. Energy gradients help define desired non-linear trajectories when point-to-point trajectories from current location to goal location are unavailable.

components of our graph using a breadth-first search. The three largest resulting strongly connected components are shown in Fig. 3.3 for the smallest object (18mm). Each grid point in this figure represents a node in the graph, and the shading of that point indicates the number of edges connected to that node. It is interesting to note that there is an extremely large connected component further out in the workspace, where most of the manipulation occurs. The next largest connected components are orders of magnitude smaller in size, and concentrated at the very bottom of the reachable workspace. Intuitively, there is no connection between these two regions. In



Fig. 3.6. Example trajectory moving the object from the left to right side of the workspace. (Top) The teal trajectory indicates the shortest path found to the goal position (green), along waypoints (pink). Throughout our progression from (a) – (d), the system determines that we have deviated off the desired path in (b), so an updated path is formulated in (c), and executed in (d) until our goal position is reached. (Bottom) Energy gradient maps are evaluated at each time step, selecting the gradient that is instantaneously closest to our desired object direction. All paths are outlined in red, whereas the green arrow indicates our desired direction and the blue arrow indicates our selected activation input.

practice, this is demonstrated very clearly by the fact that the hand cannot push the object outwards from the palm, making it a likely region for the object to become stuck. Since this is the case, path planning often fails when the object is in that region.

We also created shortest path trees between every node in the graph and all other nodes. Then, we summed the total edge distance of all shortest paths possible from a given start node. The results of this are shown as a color coding in Fig. 3.4, again for the smallest object (18mm). Essentially, the lighter colored areas in this figure are regions in the workspace that have high total edge distance sums. This means they are most connected to regions very far away, indicating a higher likelihood of finding a valid path between those nodes and randomly generated goal nodes far away. These regions occur along the lower boundary of the shaded region, along the path followed by the object when it is contacting both the proximal and distal links of the 2-link finger, as it sweeps across the workspace.

#### 3.3.1 Graph Evaluation

In this section, we evaluate properties of our constructed graph to understand connectivity between different areas of the workspace. We follow this discussion with experiments conducted on a physical robotic hand, and evaluate successful trajectories found in our implementation.

#### 3.3.2 *Experiments*

In our physical experiments, as presented in Fig. 3.6, we are able to evaluate trajectories developed by the graph and in real-time, execute computed paths. Given a valid desired trajectory, we first find the path required to reach our desired goal. Once a path is validated, we query our graph for all energy gradients for our current node, and select an actuation input that is closest to our desired next goal direction. Some goal locations cannot be reached due to friction, limited control authority, or kinematic limitations of the hand, and in these cases, a different goal position was generated. During object manipulation, it is required that we update our planned trajectories online, which is due to uncertainties in object movement and the difficultly of precisely following the original desired path. This adaptive planning approach was shown to be successful in our physical implementation, illustrated in Fig. 3.6 and further in the media attachment.

The main benefit of our energy gradient approach is that it allows us to define shortest path trajectories that are non-trivial to compute otherwise. That is, as presented in Fig. 3.5, our

computed trajectory for these two cases is not a simple point-to-point path within the workspace, but a more complex state-to-state transitions. Analyzing these paths in accordance to the geometric constraints of the gripper, we can note that sharp changes in the projected path are typically due to the fluidly changing contact scenarios throughout the manipulation.

#### 3.4 Conclusion

In this work, we presented an approach to address the problem of within-hand manipulation for caging grasps. Rather than explicitly modeling the dynamics between the object and finger contacts, we adopted the concept of energy maps to represent the underlying relationship between hand-object configurations and actuation inputs when a caging grasp is formed. By pre-computing a large set of energy maps corresponding to different actuation inputs, an energy gradient-based graph was constructed to represent the connectivity among all hand-object configurations, exploiting the local transitions enabled by the energy gradient. Using the constructed graph, we showed how to plan a sequence of actuation inputs to manipulate the object to achieve a goal state, as well as how to execute the plan adaptively to handle the uncertainties during manipulation.

In experiments, we quantitatively analyzed the properties of our proposed energy gradientbased graph for a single hand-object system, and showed that it can cover a large portion of the gripper's workspace, allowing the gripper to manipulate an object between many different positions within hand. Moreover, we showed that our approach is able to generate shortest actuation sequence trajectories for manipulation, as well as to adaptively update the execution trajectory online to improve the execution robustness. In future work, we plan to develop an objectindependent energy gradient-based map, in order to generalize the system to work with novel asymmetric objects and to change their orientation in a controlled way.

# 4 DESIGN FOR PASSIVE CAGING

Underactuated hands are particularly good at delicately grasping objects of unknown shape, size, and stiffness without complex control. This remarkable ability comes from their compliance and joint coupling—as the fingers begin to touch a target object, the forces generated at contact begin to influence the kinematic configuration of the hand. In this way, underactuated hands simply wrap around objects they are grasping, trading high forces at contact for large reconfiguration at the finger joints. But even before contact, the design features of the hand play an enormous role in determining the finger closing motions. By carefully tuning design parameters, we can produce hand closing motions that range from a stabbing-like behavior that ensures contact first occurs at the very end of the fingertips—to a passively occurring caging behavior on the other end of the spectrum. It is this idea that inspires our work, motivated also by the simplicity and accuracy of simulating this behavior. Essentially, the authors hope to show that through underactuation and intentional design choices, caging grasps can be made automatically over a wide range of objects by simply closing the hand. We hope that this work can be used as practical reference for the design of simple 'made-to-cage' robotic hands.

We narrow our scope to an underactuated hand model consisting of two 2-link fingers, each of which has a torsion spring and a pulley at each joint and a single tendon which drives both joints via their pulleys. Through simulation, we examine the effect of hand design parameters, including phalanx length, palm width, joint stiffness, and joint actuation torque on a hand's ability to create caging grasps over a wide range of object sizes and locations with respect to the hand's palm. We analyze the effect of varying these parameters on the planar caging performance for this simple underactuated hand model, considering both the workspace of object locations able to be caged as well as the "quality" of the cage, quantified by the ratio of the largest opening between



Fig. 4.1. Design parameter selection determines the largest opening between the fingers at contact, the number of reachable object positions, and in turn the passive caging ability of a simple underactuated hand.

the fingers, and the object diameter. The results lend insight into important design features of underactuated hands in terms of passively creating caging grasps.

We are not the first to consider the quality of a caging grasp. In particular, Sudsang et al. developed a cage quality metric for partial caging grasps, based on the probability of finding an escape path from a partial cage [42]. Similarly, Makita et al. developed a partial caging quality measure based on the likelihood of ejection, while considering the dynamics of the object [43]. All of this work is based on decades of research into algorithmically generating caging grasps, immobilization, and studies of geometry. Ever since Besicovitch posed the challenge problem "A net to hold a sphere" to his students in 1957, a great amount of research has focused on 'caging' grasps and developing algorithms to find them [44]. In 1990, Kuperberg posed a number of problems helping to formalize the notion of caging, which he described as "immobilizing compact sets in the plane with points" [45], [46]. Rimon and Burdick developed another theory of

immobilization and found geometric requirements for immobilizing planar objects [47], [48]. Wan et al. showed how caging could be used to guard against uncertainty during grasping [15]. Sudsang developed an algorithm capable of generating all possible caging sets for nonconvex polytopes [19], [49].

More closely related work focuses specifically on the design of hands for caging. Yoshida et al. created a quasi-static simulation of an underactuated finger closing around an object, and used it to design a two fingered underactuated gripper for more reliably grasping free-flying objects, by passively formed caging grasps [50]. Similarly, Backus designed an underactuated hand to "maximize the wrap of the digits about the object" for improving the robustness of aerial grasping and perching [51]. Other works comment on the particular ability of underactuated hands to passively create caging grasps [52], [53], [4], and many leverage design parameter optimization to produce desirable open-loop hand motions [54], [55], [32], [56], [27]. It is in this vein that the authors address this work.

#### 4.1 Underactuated Hand Simulation

We created a simple simulation to study the effect of design on the ability of underactuated hands to form caging grasps as they close around an object, placed at points on a grid in front of the hand. The following sections detail the assumptions and mathematical formulations used to model the motion of tendon driven underactuated hands, and the metrics used to quantify their ability to create caging grasps.

This work focuses on a planar two finger underactuated hand, where each finger is tendondriven and has a rigid proximal and distal phalanx. There are two single degree of freedom revolute joints on each finger, each of which has a torsional spring for passive-opening compliance and a pulley for torque transmission via a tendon. This hand model is very similar to a number of popularized underactuated hands, such as the SDM Hand [55] and the Yale OpenHand Model T42 [57]. A diagram of this hand model is shown in Fig. 4.1. We assume that our hand is symmetric, meaning the right finger is a mirror image of the left, and that the hand operates quasi-statically, meaning that it operates slowly enough that any inertial effects can be ignored. Additionally, we assume that the hand operates in position control mode. This means that an actuator position is specified, which in turn shortens or lengthens each finger tendon accordingly, while letting the torque supplied to the actuator vary.

#### 4.1.1 Underactuated Hand Model

We created a simple simulation to study the effect of design on the ability of underactuated hands to form caging grasps as they close around an object, placed at points on a grid in front of the hand. The following sections detail the assumptions and mathematical formulations used to model the motion of tendon driven underactuated hands, and the metrics used to quantify their ability to create caging grasps.

#### 4.1.2 Free-swing Trajectory Simulation

While an underactuated finger is being closed and moving freely without any external loading, it behaves as a single degree of freedom mechanism. Specifically, all joints of the finger will move simultaneously and in a deterministic way such that the fingertip follows a single path through space, known as the free-swing trajectory. Conveniently, this path can be directly determined by the ratio of the torques at the joints of the finger. To see this, we must first realize that the tension in the tendon is constant throughout its length. This means that the applied torque at each joint is determined by the product of the tension and the radius of that joint's pulley—but in order for the finger to remain in static equilibrium, this torque must be balanced by the torsional spring at that joint. Thus, using knowledge of the pulley radii and torsional spring stiffness, we

can easily compute the unloaded trajectory of an underactuated tendon-driven finger by writing an expression for moment balance about each joint.

$$\sum M = 0 \tag{1}$$

$$Tr_p - k_p (\theta_p - \theta_{rest}) = 0$$
 (2)

$$Tr_d - k_d (\theta_d - \theta_p) = 0 \tag{3}$$

 $T, r_p, r_d, k_p, k_d, \theta_p, \theta_d, \theta_{rest}$  are tendon tension, proximal and distal pulley radius, proximal and distal spring stiffness, proximal and distal joint position, and the rest position of the proximal joint. It is assumed that a physical hard stop exists that prevents the proximal joint from interfering with the palm of the hand, limiting its motion between 0 ( $\theta_{rest}$ ) and  $\pi$  radians. We also assume two hard stops on the distal joint, limiting the position to be between 0 and  $\pi/2$  radians, relative to the proximal link. Recalling that the tendon tension is the same throughout its length, it can be eliminated from these equations, yielding the coupling ratio between the joints



Fig. 4.2. In this work, caging quality is defined as the length of the smallest opening between the fingers. This can either be the distance between the fingertips, the distance between a fingertip and opposing link, or zero if the fingers interdigitate.

$$T = \frac{k_p}{r_p} (\theta_p - \theta_{rest}) \tag{4}$$

$$T = \frac{k_d}{r_d} (\theta_d - \theta_p) \tag{5}$$

$$\frac{k_p}{r_p} \left( \theta_p - \theta_{rest} \right) = \frac{k_d}{r_d} \left( \theta_d - \theta_p \right) \tag{6}$$

$$\theta_d = \frac{k_p r_d}{k_d r_p} \left( \theta_p - \theta_{rest} \right) + \theta_p \tag{7}$$

From (7), it is clear that given a proximal joint angle, a corresponding free-swing distal joint angle  $\theta_d$  can be automatically determined, based on the design parameters of the hand:  $r_p, r_d, k_p, k_d$  and  $\theta_{rest}$ . Since we know the hard stop position of the proximal joint and all of the other design parameters, we simulate a range of proximal joint angles and calculate a corresponding range of distal joint angles, based on (7). Then, we can determine the paths that the links of the finger will follow as its tendon is shortened, yielding the path of its fingertip—the free-swing trajectory. Finally, if we know the object's position relative to the hand, we can determine the configuration of the fingers at the instant contact is made, under the simplifying assumption that the fingers do not perturb the object upon contact. We find each finger configuration at contact by simulating ever-increasing proximal joint angles starting from the finger's rest position, as if

the finger is being actuated by pulling the tendon, until a link of the finger intersects the object, and contact is made.

#### 4.1.3 Cage Quality

Quantifying the quality of the cage is important because underactuated hands can reconfigure due to external disturbances, risking object ejection. A slight reconfiguration of the hand is more likely to break a looser cage than a tighter one. Thus, a tighter cage creates a more reliable grasp in the face of potential contact with other objects in the environment. In order to compare different hand designs against one another, we devised a metric based on both the tightness of caging grasps, as well as the number of object starting locations over which a caging grasp can be achieved. To construct our metric, we first determine the "quality" of a cage by computing the ratio of the largest possible opening between the fingers through which a contacted object could escape, and the diameter of the object being grasped. By this ratio, a cage of quality 1 is the minimum cage that can exist—any slight increase in the distance between the fingers breaks the cage. Conversely, a cage of quality 0 is an excellent cage—this occurs when the fingers leave no opening for the object to escape. Thus, a smaller opening between the fingers yields a higher quality cage. The length of this opening can be determined as either the distance between the fingertips, the normal distance between a fingertip and opposing link, or zero if the fingers completely cage the object via interdigitation (when the fingers mesh with one another). Examples of caging scenarios are shown in Fig. 4.2.

In addition to the quality of cage, we are also interested in the number of initial object positions from which the object can be caged by simply closing the hand, which we refer to as  $n_c$ . Ideally, we would like the initial caging capture region to be as large as possible to guarantee a successful initial cage in the face of large environmental uncertainty. Thus, we simulate grasping an object over a grid of starting positions in front of the hand (Fig. 1). Note that because our hand model is symmetric, we only need our grid to cover half of the area in front of the hand. Any result that we find for the left half will simply be mirrored on the right. Moving the object throughout our grid, we determine where it can be caged and compute the resulting quality, and count the total number of grid positions where a cage can be achieved for a given hand design. Finally, we can fully construct our metric, which we call the total cage score  $c_T$ . The total cage score for a single hand design is computed in the following way

$$c_T = \frac{1}{n_o} \sum_{n_o, n_c} (1 - c_q)$$

(8)

where  $c_q$  is the cage quality (the ratio of the largest finger opening and the diameter of the object being grasped) achieved at a single object starting point within the grid. This term is computed for each grid point where a cage can be established ( $c_q \leq 1$ ), and added to a running total for each hand for all valid caging positions  $n_c$ . Then, the metric for a single hand design is averaged over all object sizes  $n_o$ . Thus, the hand with the highest value of  $c_T$  is able to achieve a high number of quality caging grasps.

#### 4.1.4 Design Parameterization

To examine the influence of design on a hand's ability to passively create a caging grasp, we parameterized our simulated hand model so that we could vary proximal and distal joint pulley radius, proximal and distal joint spring stiffness, proximal and distal link lengths, object diameter, and palm width. In order to reduce the size of our design space to decrease computation time, we held the total finger length constant while varying the distal link length. We did the same with spring stiffness and pulley radius, holding the proximal values constant in both cases, while varying the distal values—this works because the proximal to distal ratios affect the free-swing trajectory, rather than the values of the individual components (7). Also, we assumed hand symmetry and gave both the right and left fingers identical parameters. Next, we discretized each of these parameters over a wide range of values, shown in Table 4.1.

This enabled us to create 40,000 hand designs, which we scored with our metric (8). Each hand's metric was averaged over a range of object radii (last row of Table 4.1). In this way we were able to compare the caging ability of underactuated hands over a wide swath of design space, caging a large range of object sizes. The total cage scores for 3,600 hands are shown in Fig. 4.3.

Parameter	Min	Max	Resolution
$r_p$ – proximal pulley radius	0.10		1
r <sub>a</sub> - distal pulley radius	0.075	0.15	10
$k_p$ – proximal spring stiffness	0.025		1
k <sub>d</sub> – distal spring stiffness	0.01	0.10	20
$l_p$ – proximal link length	0.10	0.90	20
l <sub>d</sub> – distal link length	$1-l_p$		20
$p_w - palm$ width	0.05	0.70	10
r <sub>o</sub> – object radius	0.15	0.30	10

TABLE 4.1 DESIGN PARAMETER VARIATION

All values are proportional to a unit length, and not physical units. In the case of stiffness, only

the proximal and distal ratio is important, so units cancel out.

### 4.2 Design Parameterization Results

Parameter selection is a delicate balancing act. As shown in Fig. 4.3, each design parameter has considerable influence over the final result. The following sections detail the effects of each specific category of design features.

# 4.2.1 Effect of link lengths and palm width

The effect of changing link lengths is most pronounced when the palm of the hand is very small. This is because when the palm is small the proximal links of the hand tend to make first



Fig. 4.3. Results of the design parameterization shows a range of caging ability (lighter is better) while varying link lengths, palm width, joint stiffness, and pulley radius. Each point represents the performance of a single hand design over its entire reachable workspace, averaged over the full range of object sizes. The lighter the color, the more high quality caging grasps a hand can make, based on the metric in (8).



Fig. 4.4. Left: The slice of design space containing the best gripper; Center & Right: The best gripper and its design parameters.

contact with the object, leaving the fingertips spread wider apart as the distal springs get stiffer. This is why the best regions of the first row of Fig. 4.3 have much lower spring stiffness ratios than the overall best results on row 2. Overall, variation in palm size produces some of the most striking differences across Fig. 4.3. The fact that row 2 contains the best results suggests that there is a palm size that is "just right" (not too large, not too small), for a fixed finger length. This is because when the palm grows too large, fixed length fingers can no longer reach objects on the far ends of the grid, reducing the number of grid positions where a cage can be achieved.

# 4.2.2 Effect of spring stiffness ratio and pulley ratio

As predicted by its role in calculating the free-swing trajectory (7), the ratio of spring stiffness is central in determining a hand's success in creating a natural caging behavior. The role of spring stiffness is perhaps most evident in the center subplot in Fig. 4.3, as it shows that improper spring selection makes the difference between some of the best grippers shown in the entire grid of subplots and some of the worst. This is because the middle column of subplots has a pulley radius ratio that is nearly equal to one. Indeed, looking at (7), when  $r_p$  and  $r_d$  are equal, the scaling of the distal joint travel relative to the proximal travel is entirely determined by the ratio of spring stiffnesses. Thus, the effect of changing spring stiffness is most amplified by similar

pulley radii. The same can be said for spring stiffness ratios near one, which increases the influence of pulley ratio. These two ratios must be chosen with care, lest they cancel each other out.

#### 4.2.3 Best gripper for caging object acquisition

The best gripper found by the search of the design space is shown in Fig. 4.4. The design parameters  $r_p$ ,  $r_d$ ,  $k_p$ ,  $k_d$ ,  $l_d$ ,  $l_p$ , and  $p_w$  are equal to 0.1, 0.075, 0.025, 0.072, 0.394, 0.605, and 0.411, normalized to a unit length or stiffness. Its distal link is shorter than the proximal link, and its palm width is approximately the same length as the distal link. Its proximal pulley is larger than its distal pulley, and it has a stiffer distal spring than its proximal spring.

#### 4.3 Conclusion and Design Principles

In this paper we examined the influence of design on the passive caging ability of underactuated hands through extensive simulation. Interpreting the results from Fig. 4.3, we can write a list of principles that can help to guide the design of future underactuated hands, for maximal caging ability.

- 1. Caging ability suffers when the palm is either too small or too large
- 2. Large palms make it more difficult for both fingers to reach objects that are off center of the hand
- 3. Small palms result in proximal-link-first contact, pushing the object away
- 4. In the case of a small palm, special attention should be given to both the pulley radius ratio and stiffness ratio to ensure the distal links close in quickly enough to cage the object
- 5. Pulley ratio and stiffness ratio complement each other—and the effect of either is amplified when the other is close to unity (see (7))

In short, if you are designing a hand for better caging performance, select design parameters that follow these principles, or are found in a bright region of Fig. 4.3 for best results.

# 5 ENERGY MODEL DESIGN FOR PLANAR CAGING MANIPULATION

Humans use all surfaces of the hand for contact-rich manipulation, in contrast to robot hands which typically utilize only the fingertips, often limiting dexterity. In this work, we leverage a potential energy-based whole-hand manipulation model that does not depend on contact wrench modeling like traditional approaches, and use it to design a robotic manipulator. Inspired by the loose caging grasps and full dexterity observed in human manipulation, a metric is developed and used in conjunction with the manipulation model to design a two-fingered dexterous hand, the Model W. This is accomplished by simulating all planar finger topologies comprised of open kinematic chains of up to three serial revolute and prismatic joints, and evaluating their performing continuous object reorientation, as well as repeatedly alternating between power and pinch grasps – two contact-rich skills that have often eluded robotic hands – and we experimentally characterize the hand's manipulation capability. This hand realizes manipulation motions reminiscent of thumb-index finger manipulative movement in humans, and its topology provides the foundation for a general purpose dexterous robot hand.

The dexterity of a robotic hand can be greatly increased when all of its surfaces are used for manipulation, rather than just the fingertips [12]. Despite this, the majority of manipulation research is rooted in the assumption of fixed contact points between the tips of the fingers and the object, a precedent established many decades ago [58, 59]. This is in part due to the complexity involved in modeling and controlling non-fixed contacts, which would be necessary to prevent object ejection during force closure-based rolling or sliding fingertip manipulation. Accurately predicting the motion of sliding contacts depends on good knowledge of both force magnitude and direction at the point of contact, in addition to the coefficient of friction between the object and the fingers [60, 61, 62], which changes over time [63]. Predicting the motion of rolling contacts similarly depends on these properties, as well as good knowledge of local surface geometries [64, 65, 66]. Many of these parameters are challenging to measure a priori, and even more difficult to measure in real time with novel objects in a changing environment, though some promising sensor advances have been made [67].

In this work, we demonstrate highly dexterous manipulation that utilizes all of the hand's internal surfaces, by a hand that was specifically designed for this purpose. This hand enables manipulation where contacts can seamlessly shift between rolling, fixed, and sliding modes, while object ejection is prevented through caging, and its control does not depend on knowledge of contact forces, friction coefficients, or local geometries. The work was motivated by observing human manipulation, which is highly dexterous and utilizes all surfaces of the fingers and palm, without enforcing fixed contact. With the presented work, we make progress towards that kind of dexterity in two main ways. First, we treat the hand-object-system holistically and utilize the observation that the system can be viewed in terms of total summed actuation effort and overall system energy, with corresponding variations in energy based on object location and configuration. Instead of the traditional method of calculating and controlling individual joints and actuators to result in a desired overall system configuration, we drive grasped objects to desired configurations by varying the potential energy in the system, forcing objects to follow energy gradients to a new

state. This is accomplished without having to precisely model contacts or wrenches within the system, and is robust to the uncertainties that typically make producing controlled sliding or rolling extremely difficult. Second, we perform these manipulative actions while ensuring that the fingers "cage" the object [16, 2, 3]. This physically prevents the object from being ejected during manipulative movements that are generally risky to perform, such as stick/slip transitions.

Many robotic manipulators have been designed to perform specific dexterous within-hand manipulation tasks, such as object reorientation [68, 32], object translation [69], or both [70, 71]. At the core of every one of these designs is a Forward Motion Model (FMM), a mathematical relationship between actuator inputs and object motions. The FMM enables a prediction of object motion, given a controllable feature of the system, such as actuator torque or position. Using this prediction, it is then possible to change design parameters within the system and observe changes in the object motion, until a desired behavior is reached. In some cases, the FMM is purely kinematic [29, 41], in others detailed modelling of contact modes (e.g. rolling, sliding, fixed, breaking) is included [72], and some even model inertial forces and dynamics [8].

In this work, we use a potential energy-based FMM that we previously introduced for use with caging-based manipulation with underactuated hands [4, 73, 5]. This FMM, known herein as the Energy Model, does not require direct consideration of force-closure or grasp stability, and is formulated for the first time in this work as a constraint based minimization. It was inspired by simple linkage-based kinematic gripper models, and the energy-based analysis of grasping ability found in [20, 74]. A basic model of caging [6] was enforced to prevent object ejection, and was combined with the object motions predicted by the FMM to form a manipulation metric. This metric was then used to assess the performance of different symmetric hand designs based on every planar open kinematic serial chain finger, made of up to three revolute and prismatic actuators.

Finally, after the design space search yielded a highly dexterous robotic hand that can manipulate objects of many different sizes without ejection, it was constructed and its dexterity was demonstrated. This hand, called the Model W and shown in Figure 1, has a prismatic palm and two serial revolute joints per finger, and its motion unintentionally resembles that of thumb-index finger manipulative movements in humans, hinting at a potential design strategy for future robotic manipulators.

#### 5.1 Energy-based Model Formulation

In this section we describe the formulation of our energy-based forward motion model. In short, we compute a hand-object workspace, such that each point in the workspace has an associated energy value. We term this workspace an energy map. There is one energy map (in the form of a three-dimensional scalar field where the dimensions represent the object's planar position and orientation) associated with each commanded actuation input to the system. We compute the gradient of each energy map, resulting in a vector field for each actuation input. Each vector in the field corresponds to the direction and magnitude of motion that can be realized by the object at that configuration in the workspace, given the corresponding actuation input. Thus, for a given object position and orientation in the workspace of the hand, there will be many distinct directions of feasible object motion, corresponding to the collection of all gradient vectors from all possible actuation inputs to the system.

#### 5.1.1 Forward Motion Model

This theory is based on the idea that with enough mechanical work, a position controlled motor can be displaced from its commanded set point. Equivalent in magnitude to the work put into the motor, the potential energy gained by a motor during such a displacement is

$$U = \tau_m(\alpha - \theta) \tag{1}$$

52
where  $\theta$  is the commanded set point,  $\tau_m$  is the torque generated by the motor (assumed to be a source of constant torque), and  $\alpha$  is the new displaced position of its shaft. A displacement can be caused by an external disturbance that cannot be resisted by the torque generated by the motor. This displacement may occur on the output of a transmission, such that it is propagated backwards to the motor. In the case of a transmission, we can relate the observed output displacement to the motor shaft displacement by  $q = K(\alpha - \theta)$  where K is the transmission ratio,  $\alpha - \theta$  is the observed displacement of the output, and q is the displacement of the motor shaft. If multiple actuators are displaced, their respective energies are simply summed to compute the system's total energy. The total increase in a system's potential energy due to a displacement of multiple actuators with unique transmissions is

$$U = \sum_{i=1}^{n} \tau_i K_i (\alpha_i - \theta_i)$$
<sup>(2)</sup>

where n is the number of actuators in the system. As an example, imagine a simple planar finger modeled as a two-link revolute serial chain, with a motor at each joint. If each joint of the finger is commanded to a known set point, and an object is forcibly brought past the point of contact with both links of the finger, both motors can be backdriven simultaneously and both joints will be displaced, due to the work done on the finger by the forcible placement of the object.

#### 5.1.2 System Kinematics

This displacement in motor position can easily be mapped to displacements in Cartesian positions of joints of the finger using forward kinematics transformation matrices, constructed from Denavit-Hartenberg parameters. The Cartesian position of a frame affixed to joint *k* of a *n* link serial chain linkage with respect to a fixed global frame is  $u_k$  extracted from the transformation matrix  ${}^{0}T_{k}$  where

$${}^{0}T_{k} = \begin{bmatrix} R_{k} & \boldsymbol{u}_{k} \\ 0 & 1 \end{bmatrix} = \prod_{i=1}^{k} {}^{i-1}T_{i}(\theta_{i}), \quad k \leq n, \quad \boldsymbol{u}_{k} \in \mathbb{R}^{N}$$
(3)

such that  $R_k$  is the rotation matrix associated with the *k*th joint,  $u_k$  is the joint position vector for the *k*th joint, and  ${}^{i-1}T_i(\theta_i)$  is the homogeneous transformation matrix that transforms points from the link i - 1 affixed frame to link *i*. We assume that joint affixed frames follow the standard Denavit-Hartenberg convention, with the *z*-axis aligned with the joint axis, and we also set positive joint displacements such that they result in a finger becoming more closed. For each finger, the set of all joint positions at the commanded actuation input  $\theta$  is *P* such that

$${}^{0}P_{\theta} = \left\{ u_{\theta,k} \middle| k = 1, \dots, n \right\}, \boldsymbol{u}_{k} \in \mathbb{R}^{2}.$$

$$\tag{4}$$

An object in pose q can be represented by a set S of m boundary points with respect to a fixed global frame 0 such that

$${}^{0}S_{q} = \{s_{q,j} | j = 1, ..., m\}, s_{j} \in \mathbb{R}^{2}$$
(5)

where  $s_i \in \mathbb{R}^2$  for planar objects.

We imagine an object fixed in space displacing the link of a finger by a known amount in Cartesian space, but we actually wish to know the displacement of the motor driving that link of the finger, in order to compute the gained potential energy. This implies the use of inverse kinematics—to find a joint angle that produces a desired link position, such that contact occurs with the object somewhere along the finger. However, because we are not limiting this work to fingertip manipulation, we do not know precisely where contact occurs between the link and the object, and instead of directly calculating candidate joint positions using inverse kinematics, we frame the problem of determining displaced motor positions as a mathematical program that minimizes the system's total energy, utilizing the homogeneous transformation matrices from Equation 3 in a nonlinear constraint. Specifically, we formulate a constraint that does not allow

any part of a finger to penetrate an object, or pass from its open configuration through the object to the other side.



Fig. 5.1. Manipulability is derived from the convex hull formed by gradient vectors of the system's potential energy scalar fields. (A) Each contour plot shows the potential energy for a fixed square object configuration and a symmetric *RR* hand, given a distinct set of actuation inputs. The dashed lines represent the commanded finger positions, given the actuation inputs for each joint (in radians) listed below each contour plot, and the solid lines represent the realized finger positions, were they to close around the object in its fixed position. The red vectors emanating from the center of the object represent the gradients of the potential energy fields at the object's center point. The yellow vectors are the red vector scaled by the strength of the basic cage created around the object by the fingertips, a value from 0 (no cage) to 1 (fingers interdigitate). (B) The manipulation metrics are calculated by joining the tails of all vectors, calculating the convex hull of their tips, and finding the radius of the largest origincentered ball contained within their hull. Concretely, this radius is proportional to the minimum wrench the fingers can apply to the object in any direction (in the object's configuration space  $x-y-\beta$ ). (C) The manipulability (without caging) is represented by the radius of the red ball centered at the origin, and (D) the caging manipulability by the radius of the yellow ball.

Since we are interested in finding the displaced configuration of the system that corresponds to its lowest energy state, we formulate an optimization problem to minimize the energy. To compute the potential energy gained by a position controlled serial chain finger that is displaced by an object of known geometry and configuration, we compute the minimal potential energy state of the system according to the following constraints

$$U_{\theta,q}^{*} = \min_{\alpha} \sum_{i=1}^{n} \tau_{i} K_{i}(\alpha_{i} - \theta_{i})$$
(6)
  
s.t.  $0 \le \alpha_{i} \le \theta_{i}$ 
 $u_{x} s_{y} - u_{y} s_{x} \le 0, \forall s \in S_{q}, \forall u \in P_{\alpha}$ 

where  $U_{\theta,q}^*$  is the minimum system energy for object pose q and commanded actuation input  $\theta$ . The displaced hand configuration corresponding to the minimum energy of the system is represented by  $\alpha^*$ ,  $u_x$ ,  $u_y$ ,  $s_x$  and  $s_y$  are the x and y components of the u and s vectors (joint positions and object points),  $P_{\alpha}$  is the set of all joint positions from actuation input  $\alpha$ , and  $\theta_i$  is the commanded set point of the *i*th joint. The constraints  $\alpha_i \ge 0$  and  $\alpha_i \le \theta_i$  represent that the displaced actuator positions must be greater than or equal to zero, usually due to mechanical hard stops, and must be less than or equal to the commanded set points  $\theta$  since the object cannot pull the finger past where it is commanded to go. The final constraint  $u_x s_y - u_y s_x \le 0$  accounts for which side of the object the finger is on, and is derived from the sign of the cross product of two vectors originating at a joint, with one directed to the subsequent joint (or fingertip) and the other to a point on the object. When the sign of this constraint drops below zero, it is physically equivalent to the finger passing through this point on the object, or having passed completely through the object to the other side. Visually, this occurs when the tip of one vector passes across the other (remembering that their tails are connected). Using the logic from Equation 2, we can simply solve the mathematical program in Equation 6 once for each finger and sum their minimum system energies to obtain the total hand-object potential energy.

#### 5.1.4 Energy Fields

The solution to the energy minimization yields the displaced, minimum energy state of the system corresponding to one single pose of the object, given commanded finger actuation inputs  $\alpha$  and fixed object pose q. This can be extended to the set of all possible poses of the object for that actuation input,

$$U_{\alpha} = \left\{ U_{\alpha,q}^* \middle| q = 1, \dots, Q \right\}$$

$$\tag{7}$$

resulting in a scalar field  $U_{\alpha}$  that, when visualized, lends good intuition about how the hand will move the object. In addition to considering how the hand is displaced by the forcible placement of the object in a specific pose, we can also consider how the object will be displaced by the actuation of the hand, should we drop our assumption that it is fixed firmly in place. As an example, a hand's translation capability with a square object is illustrated by the contour plots showing the scalar field  $U_{\alpha}$  in Figure 5.1, where the contour color is the potential energy magnitude at that specific position in the hand-object workspace. Whereas we were initially only concerned with the lowest energy configuration of a hand, we now imagine that the object is released from its fixed configuration and free to move along with the hand, and that the entire system will settle to its lowest energy configuration. From the contour map, it is clear to see where the object will tend to move—toward the lowest energy region in the hand's reachable workspace.

## 5.1.5 Gradient of Energy Map

The gradient of the potential energy scalar field with respect to the object's planar pose coordinates x, y, and  $\beta$  results in a distinct vector field

$$\gamma_{\theta} = -\nabla_{x,y,\beta} \mathbf{U}_{\alpha} \tag{8}$$

for each scalar field  $\mathbf{U}_{\alpha}$  corresponding to a distinct actuation input  $\alpha$ . Each vector field  $\gamma_{\theta}$  consists of pose q dependent planar motion vectors  $w_q \in \mathbb{R}^3$  of the form  $w_q = [dx_q, dy_q, d\beta_q]$ . Each of these vectors is the net wrench that can be imparted on the object, or more generally the motion that can be realized by the object at pose q given the actuation inputs  $\alpha$  from the hand. The set of all vector fields  $\Gamma = {\gamma_1, ..., \gamma_{\Theta}}$  can be used to evaluate manipulation capability throughout the workspace of a hand-object system.

Specifically, if we consider a single object pose q with respect to the hand, we can collect the set of vectors  $\mathbf{W}_q = \{w_{q,1}, ..., w_{q,\Theta}\}$  (one vector from each field  $\gamma_{\theta}$ ), where each vector corresponds to a possible actuation input to the hand while the object is at pose q. The span of this set of vectors in  $x - y - \beta$  space tells us how well the hand can manipulate the object from its pose in the workspace and is at the core of the manipulation metric used in this work.

#### 5.1.6 Manipulation Metric

Given the set of net wrench vectors  $\mathbf{W}_q$  that can be imparted on an object at a known pose, we can determine the possible directions of motion that can be realized by the object. This is illustrated visually by the sets of red and yellow vectors in Figure 5.1. The convex hull of  $\{w_{q,1}, \dots, w_{q,\Theta}\}$  is  $Conv(\mathbf{W}_q)$ , and is the set of all allowable configurations of net wrench application to the object. We want to find the maximum radius ball that can be inscribed in  $Conv(\mathbf{W}_q)$ , since the radius of this ball is equivalent to the maximum net wrench magnitude that can be applied to the object in any direction.

In other words, the radius of this ball allows us to compare how well the hand can manipulate the object in one pose, relative to other poses. The greater the radius, the larger the wrench we can exert on the object in any direction. To compute the maximum radius ball, we use the hyperplane representation of a polytope formed by  $Conv(\mathbf{W}_q)$ . Recall that any polytope is

defined by the intersection of hyperplanes  $\langle h_i, w \rangle \leq b_i$ . See [75] for more of this theory, and [76] for the theory behind producing any polytope representation given vertices, as we have in  $Conv(\mathbf{W}_q)$ . Finally, given a hyperplane representation, note that the radius  $w_q^*$  of the maximum radius ball may be computed with the quadratic program

$$w_{\mathbf{q}}^{*} = \max_{w} ||w||^{2}$$

$$s.t. \quad \langle h_{1}, w \rangle \leq b_{1}$$

$$\langle h_{2}, w \rangle \leq b_{2}$$

$$\vdots$$

$$\langle h_{n}, w \rangle \leq b_{n}.$$

$$(9)$$

A visual example of the largest origin-centered ball bounded by the convex hull of the gradient vectors is shown by the red and yellow balls in Figure 5.1.

#### 5.1.7 Caging Metric

While a larger radius tells us that we have greater control authority over the object at a given pose, it tells us nothing about the likelihood of object ejection while continuing manipulation in a given direction. Ideally, we would like to design a hand that can safely manipulate an object in any direction from any position in the workspace, while safeguarding against object ejection. To quantitatively consider this tradeoff between manipulability and ejection prevention, we scale each of the vectors in  $\mathbf{W}_q$  by a distinct value between 0 and 1, derived from a basic representation of how caged the object is, given the hand's configuration under actuation input  $\theta$ . If the hand does not cage the object at all (meaning that the smallest distance between the fingers is larger than the object itself), the vector is multiplied by 0, meaning it is not safe to proceed actuating in that direction. If the fingers completely cage the object (meaning there is no gap between the fingers), the vector is multiplied by 1. When a gap exists between the fingers, but it is smaller than the

diameter of the object, the vector is scaled by a value between 0 and 1 based on the size of the gap normalized by the size of the object. Now, using this set of scaled vectors, the same procedure is used as in Equation 9 to find the largest inscribed ball inside of the caging convex hull, resulting in the maximum caging manipulation radius  $\mathbf{w}_{cq}^*$  and illustrated by the yellow vectors and ball in Figure 3. The caging manipulability score  $H_0$  for a hand-object system is then

$$H_O = \frac{1}{U_{max}Q} \sum_{q=1}^{Q} \mathbf{w}_{cq}^*$$
(10)

where O is the object being manipulated and  $U_{max}$  is the maximum potential energy the system could theoretically see given the actuation limits of the hand. This acts as a scaling factor that accounts for the fact that gradient vectors grow longer (and thus the radius of the largest ball increases) as you add more actuators to the system (you are able to exert a larger wrench on the object given more actuation energy). In simple terms, H is just the average caging manipulability that can be had per actuation-workspace-unit. Special care was taken to ensure the final design space results were not skewed towards hands with small reachable workspaces (small Q values), as very high values of H can be obtained when Q is small. The overall score for a hand design is the average score  $H_0$  over all simulated objects, as shown in Figure 2 (written simply as H, as in the colorbar label).

#### 5.1.8 Kinematic Topology Enumeration

Using the previously described manipulation metrics, this work compares hand designs consisting of two opposing symmetric fingers, each modeled as an open serial chain linkage. In this section we enumerate all possible planar serial chain finger topologies consisting of up to three revolute and prismatic joints. We show why it is unnecessary to consider more than three joints per finger in the plane, and select the best candidates for further consideration. We begin by limiting the scope to open serial chains, meaning that we are interested in mechanisms where one end is fixed and the opposite end is free, and all joints are in series with one another. While it is likely that fingers based on parallel mechanisms would create highly dexterous hands, we choose to save their consideration for future work, allowing us to consider a tractable space of kinematic topologies herein. Specifically, we focus on planar open serial chain linkages comprised of revolute (*R*) and prismatic (*P*) joints, separated by rigid links, where each linkage has a fixed base link (the palm). The number of possible finger topologies for a finger with *n* actuators is  $2^n$ , and is found by counting the number of ordered samples with replacement for two actuation styles (*R* & *P*). The full list of finger topologies considered in this work is *R*, *P*, *RR*, *PP*, *RR*, *PPP*, *RRP*, *PRP*, *RPP*, *RPP*, *RPP*, and *PRR*.

## 5.1.9 Design Space Variation

To determine how design can be leveraged for better manipulation, 6,250 unique hand designs were simulated manipulating 10 distinct objects over their entire workspace, and the manipulation ability of each hand-object system was graded based on the metric described by Equation 10. All topologies enumerated in the previous section were simulated. Design parameters including the palm width, distal-most link length, and prismatic joint orientation with respect to the previous link were varied.

Key design parameters were varied in order to exhaustively explore the design space of planar open serial chain fingers. Specifically, the palm width p (the spacing between the fingers), the distal-most link length d, and the angle of prismatic joints with respect to the previous link  $\phi$  (typically the  $\theta$  in Denavit-Hartenberg parameter) were all varied, as shown in Figure 5.2. Each



Fig. 5.2. Simulation design parameters, visualized. (A) Two example topologies *PPR* and *R* are shown here with labeled dimensions. The distal-most link length is *d*, the angle of prismatic joints is  $\phi$ , middle links are of length *l*, and the palm width is *p* (one half of the palm is considered to be part of each finger). (B) The ten simulated objects are shown here at the same scale as the hand and grid in (C). Five distinct sizes of square objects and circular objects were simulated. The square objects were simulated at six orientations each, to capture the hand's ability to reorient them. (C) The objects were simulated at all positions shown in the 24x12 grid (example *RR* hand is shown for scale).

finger length (including half of the palm) was set to a dimensionless value of 1, so the values of these parameters can be thought of as proportions of total finger length, rather than absolute values, permitting straightforward scaling of the system according to expected object sizes. In general, the finger link lengths were determined as follows. First, half of the palm width and the distal-most link length were subtracted from the overall finger length of 1. Next, the remaining portion of the finger was divided into n - 1 equal parts, where n is the number of actuators in the finger. Some special cases exist, for example in the case of topologies containing only one actuator, the distal link length is set to zero, and length of the single link is 1 - p. In addition, when the first joint in

a finger is prismatic, the length of its link is set equal to half the palm width—in other words, a proximally located prismatic link is considered to be part of the palm. Mathematically, the assignment of link lengths can be expressed as

$$l = \begin{cases} 1 - p, & n = 1\\ 1 - p - d, & n > 1 \end{cases}$$
(11)

where p is half of the palm width, d is the distal link length, and l is the length of all other links. In practice, a palm width and a distal link length are prescribed and the remaining links lengths are solved for using Equation 11. The parameter p was varied from 0 to 0.5 in 5 steps, and the parameter d was varied from 0.0625 to 0.4 in 5 steps. As stated, the fixed angle of prismatic joints was varied from 0 to  $\pi/2$ , as shown in Figure 2. This is equivalent to varying the Denavit-Hartenberg parameter normally represented by  $\theta$ , which is the controllable parameter of a revolute joint, but remains fixed for a prismatic joint. Also, note that the transmission ratio K is different for R and P actuators. This is due to the gearing required to produce the linearly actuated prismatic joints, using standard rotary motors. In this case, we assume prismatic joints are realized with a rack and pinion, and that the pinion radius is 0.15 (again a proportion of overall finger length), but other values can be considered depending on design constraints. This value was chosen to approximately adjust each pinion motor's backdrivable range with that of the other motors located at revolute joints (so the pinion motor has approximately  $\pi/2$  rad of usable range while running the carriage over the desired distance on the linear rail) to equalize the energy cost of displacing both *P* and *R* joints.

## 5.1.10 Object Size, Shape, and Pose Variation

Each hand was simulated while varying the pose, shape, and size of a target object. Specifically, each hand was simulated manipulating ten distinct objects—five circles and five squares. Each shape's radius was varied over five sizes, from 15% to 40% of the total finger length. The pose of each object was varied to determine manipulability over the hand's entire workspace. Since the system is planar, the pose consists of an *x*-coordinate and a *y*-coordinate of the object's center, and an orientation  $\beta$ . For circular objects, orientation was not varied. The metric detailed in Equation 10 was computed at each valid pose for each object.

A valid pose of the object is one that is reachable by both fingers of the hand. If contact cannot be made between the hand and the object in a particular pose, the metric was not evaluated, as that pose is not within the hand-object system's reachable workspace. The range of simulated object sizes is shown in Figure 5.2. While the *y*-coordinate of the object pose was allowed to extend to 0, in practice those *y*-coordinate values smaller than the object radius were excluded from the grid, as their inclusion would place the object in intersection with the palm. The grid sampled in the simulation is also shown in Figure 8.

#### 5.1.11 Simulated Actuation

Joint limits were set for each joint, the range between these limits was discretized into a set of individual set points for each joint, and the Cartesian product of all sets was calculated to create the set of all possible combinations of actuation inputs for a particular finger. The actuation limits for each joint were chosen based on typical limits seen in robotic hands. For instance, the first revolute joint in a hand has a range between 0 and  $\pi$  (where both fingers at 0 corresponds to the hand being fully open), allowing it to sweep from one side of the palm to the other. Successive revolute joints however, were limited to a range between 0 and  $\pi/2$  with respect to the previous link to prevent excessive collision with other parts of the hand, or hyperextension. Prismatic joints ranges were also selected based on practical considerations, such as the approximate doubling in length that can be achieved with standard leadscrew driven linear actuators. Thus, their actuation was limited between l/2 and l. However, prismatic joints located proximally in the finger were



Fig. 5.3. Simulation results from the design space search. Hand topologies are represented by simple models, where revolute joints are cylindrical and prismatic joints are rectangular prisms. (A) The *RR* topology is shown alongside the manipulability (*H* from Equation 10) scale, which goes from 0 (cannot manipulate any object in all directions) to 1 (best manipulability for all objects). (B) The manipulability design space (from convex hull of gradient vectors) for the *RR* topology, where each shaded pixel represents a different hand design consisting of parameters *p* (palm width), *d* (distal link length), and  $\phi$  (fixed prismatic joint angle). (C) The caging manipulability design space (convex hull of modified gradient vectors). See the section Energy Fields for more details. (D-L) The caging manipulability for all other viable topologies. The best design is in  $\phi = 0$  for *PRR*. (L) The *y*-axis is  $\phi$  instead of *d* in this plot only. (M) Topologies that were not capable of fully manipulating any objects.

allowed to extend from 0 to l since clever packaging is possible within the palm. Special care was

taken to ensure that this treatment did not skew the results towards kinematic topologies with proximally located prismatic joints, by testing with and without this condition, and by ensuring that results were normalized by workspace size and did not tend towards hand designs with large proximal prismatic links.

5.2 Results

## 5.2.1 Design Space Search

A total of 6,250 unique hand designs based on the 14 kinematic topologies detailed in the section Kinematic Topology Enumeration were simulated manipulating the ten distinct objects described in the section Object Size, Shape, and Pose Variation, and their caging manipulation ability was graded using the metric described Equation 10. The simulation was parallelized in MATLAB, and run on 640 cores of the Yale High Performance Computing resource with a total runtime under 12 hours. Selected design spaces are visualized for these topologies in Figure 5.3. The results match well with intuition, as simple hands such as the *R* topology are incapable of fully manipulating objects in all directions, since they can either push or pull objects with respect to the palm depending on their size, but cannot do both from any point within the workspace. Hands based on the *RRR* topology perform well for small palm widths, as having a smaller palm than the smallest object you intend to manipulate allows you to push it away from the palm, while the fingers are dexterous enough to shift objects in all other directions.

Out of all unique hands sampled, a few stood out from the rest—capable of caging many objects while achieving high levels of dexterity throughout a large workspace. Specifically, a variant of the *PRR* topology with design parameters of  $\phi$ , *p*, *l*, and *d* equal to 0rad, 0.25, 0.43,



Fig. 5.4. The physical hand model with labels, and a human thumb-index finger manipulative motion for comparison. (A) CAD renderings of the Model W in different palm configurations.(B) The spreading of the Model W's palm is similar to the ability of the human hand to spread the index finger and thumb. (C) The physical Model W, with six Dynamixel XL-320 servos. *R* stands for revolute joint, *P* stands for prismatic joint.

and 0.32 respectively stood out with the highest manipulability score across all simulated object geometries. As described in the section Design Space Variation, these values are proportions of total finger length, which includes half of the palm, and can be scaled to any desired physical dimension.

## 5.2.2 The Model W Hand

The physical hand was designed from the optimal design topology and parameters resulting from the brute force search of the design space. The hand was designed to utilize inexpensive components and a simple design, and its design will soon be released through Yale OpenHand (an open-source robot hand hardware initiative) in hopes of making it a useful tool to others in the manipulation community. From proximal joint to fingertip, its finger length is 108mm, and the space between its proximal joints (its palm) can expand from 0mm to 72mm. It is actuated by 6



Fig. 5.5. The Model W has a large fully connected workspace, and can continuously rotate asymmetric objects. (A) Shows the representative workspaces with the four test objects shown in (C). (B) Shows the hand's ability to translate objects in every direction. Each photo shows the hand manipulating the T3 object *from* the diagonally listed direction in its row, and *to* the direction in its column. For example, the photo in the first row and second column shows the hand manipulating *from* a power grasp *to* a pinch grasp. (C) The test objects used during benchtop experiments. (D) An example of the hand continuously rotating the square test object. (E) Data from the random orientation goal servoing showing that the hand can orient the square object at any orientation.

Dynamixel XL-320 servos, which are among the least expensive commercially available smart servos—an order of magnitude lower in cost than typical Dynamixel servos. All parts were either commercial off the shelf (COTS), 3D printed from ABS on a Stratasys uPrint, or cut from acrylic using a laser cutter. Each finger is comprised of two serial revolute joints connected to a carriage

that translate on a linear rail. The distal links of the fingers were made to interdigitate, to better approximate the simulated hands. The final hand design is shown in Figure 5.4. One of the most challenging aspects of physically designing this hand was allowing the proximal finger joints to come together coaxially (to achieve a zero palm width configuration), with their axis coplanar with the palm. This was achieved using a thin floating palm and cantilevered proximal joints, as shown in Figure 3.

## 5.2.3 Workspace Evaluation with Test Objects

The hand's manipulation capability was first evaluated using 3D printed test objects. To initially assess the hand, basic grasps (power, pinch, left, right) were pre-determined for the test objects and manipulation was achieved by switching between them in an open-loop fashion. Surprisingly, the hand was able to perform these motions seamlessly at very high speeds, without ejecting the object, likely due to the inclusion of caging in the metric. Next, a simple controller was implemented that interpolated between these grasps, based on a simple visual servoing-based controller, while limiting the torque output of each motor. Limiting the torque allowed the motors to be driven to stall without worry of overheating, enabling manipulation of objects of all sizes with the same set of pre-determined grasps.

Using this controller, objects were manipulated to randomly generated waypoints in front of the hand, and the system was allowed to run continuously until 150 waypoints were reached, per object. The resulting workspaces of four test objects are shown in Figure 5.5. The controllable translational workspaces are shown for all objects, and an additional plot showing the controllable rotational workspace is provided for a square object. The workspaces are large compared to the hand, most notably along the axis normal to the palm. This is partly because the hand is able to perform a power-to-pinch transition, a challenging task for many prehensile manipulators. The



Fig. 5.6. The Model W can reorient and perform pinch to power transitions with real world objects. (A) A bottle of mustard is reoriented on a table, grasped, and squeezed. (B) A Rubik's cube is rotated continuously within the hand. (C) An orange is transitioned from pinch grasp, to power grasp, and back again. \*The background of all photographs was darkened to better highlight the hand-object system.

rotational workspace shows that the hand is able to continuously reorient the square object within the hand, another task that is often challenging for prehensile hands.

## 5.2.4 Real World Manipulation Scenario

The Model W was fixtured to a 7 Degree of Freedom (DOF) WAM® Arm (Barrett Technology) and a variety of manipulation tasks were performed using the Yale-CMU-Berkeley (YCB) Object Set [77] to demonstrate its performance in real world tasks. First, the hand was used to grasp different objects, showing the range of object sizes that can be effectively picked by the hand. The hand can easily grasp very small items, like dice, as well as very large objects like the



Fig. 5.7. The Model W can manipulate multiple objects at once, and can be controlled using teleoperation. (A) A golf ball (B1) and a squash ball (B2) are rotated about a common point of rotation. (B) Two Chinese Baoding Balls (B1 and B2) are rotated about a common point of rotation within the hand while it is simultaneously moved to different waypoints within the plane, demonstrating the how caging can prevent object ejection during manipulation with external disturbances. (C) The hand manipulates different wooden blocks into holes of matching shapes using teleoperated control. \*The background of all photographs was darkened to better highlight the hand-object system.

wider dimension of a Cheez-It box or a mini soccer ball. The range of graspable objects is largely due to the prismatic nature of the palm, allowing the fingertips to come completely together or to spread apart over 12 inches.

Next, a series of open-loop manipulation tasks were performed to demonstrate within-hand manipulation of real world objects. As a first manipulation task, both rigid and soft

cubes were continuously reoriented on a support surface. The first one, the Rubik's cube from the YCB object set, was easily reoriented using a sequence of preprogrammed grasps. The second object, a hand-made deformable knit cube (50mm) containing beans, was also successfully reoriented in a continuous fashion, albeit much slower because of less efficient force transmission due to deformation. The next manipulation scenario performed was to reorient, grasp, and squeeze a mustard bottle. The bottle, which was placed on a surface in front of the arm, was rotated approximately 90 degrees within-hand to facilitate a proper squeeze, grasped and lifted above a plate, and finally mustard was squeezed onto a plate. Following this task, the plastic YCB orange was repeatedly shifted from power to pinch grasps, and vice versa. Photographs taken during the completion of these tasks are shown in Figure 5.6.

A series of teleoperated tasks were also performed using a Shape Sorting Cube toy, motivating hand dexterity over arm dexterity, and demonstrating within-hand manipulation and controllability, despite the rather high number of individual actuators in the hand. During these tests, the hand successfully and repeatedly manipulated a set of blocks into their respective holes.

Finally, the hand was tasked with manipulating multiple freeform objects at once, to assess its ability to handle unstructured objects. First, we successfully manipulated both a squash ball and a golf ball simultaneously, juggling them around a common center of rotation within the hand. Not only are the balls different sizes, but their surface textures are very different from each other, the golf ball being hard and dimpled, and the squash ball being rubbery and compliant. Even with these differences, the hand was easily able to perform this task in both directions (CW and CCW juggling around a common center of rotation). As a final appraisal of the hand's abilities, we manipulated Chinese Baoding Balls, which are sometimes used to improve human dexterity following surgery. Using the same controller from the squash / golf ball manipulation, the hand was once again able to juggle these balls, even though they are much larger in diameter. We then programmed the robotic arm to continually travel to different waypoints in the plane while performing this manipulation, applying constant external disturbances to the balls. Despite the added disturbances to the system, the manipulation continued successfully and the balls were not lost from the grasp. Indeed, by creating a cage during manipulation, objects are much less susceptible to being ejected from the hand, and their stability in the grasp is no longer dependent on parameters related to contact position, forces, or mode. Still photographs from these experiments are shown in Figure 5.7. All of these demos and more are shown in the accompanying Main Text Video.

#### 5.3 Conclusion

## 5.3.1 Design Space Search

The results in Figure 5.3 highlight a number of key takeaways that can be used as general guidelines for the design of future manipulators. First, hands with proximal revolute joints perform best with small palm widths, as illustrated by topologies *R*, *RR*, *RP*, *RRR*, *RPP*, and *RRP*. This is because these topologies are unable to perform power-to-pinch transitions with objects smaller than their palm width. Conversely, topologies with prismatic palms avoid this problem, as they are all capable of a wide range of palm widths. This explains the stark visual difference between the best performing topology *PRR* which has a fully shaded design space, and others beginning with a revolute joint that only have a shaded vertical bar for small palm width designs. Next, fully symmetric prismatic topologies are poorly performing in part due to their inability to easily reach each finger across the palm, limiting the shared workspace of both fingers, and also because they can only apply motion in a fixed direction. Intuitively, it would seem that *PPP* should be able to manipulate very well. However, high quality manipulation is only possible with

asymmetric designs of this topology, as you need one finger to push the object away from the palm, while the other is able to draw it closer—one finger is not capable of both since frictional forces are not considered in this model.

Another takeaway is that modifying the metric to include caging moderately lowers the scores of many hands. In other words, the design space shown in Figure 5.2 is perhaps less intuitive than the corresponding design space that does not include caging. Many topologies are actually better at manipulation than illustrated by the plots, but simply cannot manipulate while simultaneously caging the object. This is shown by the comparison in the box containing the *RR* topology, where the plot corresponding to 'no caging' has more pronounced shading than 'caging'. When considering hand designs based on manipulation ability alone, both *PRR* and *RPR* were top performers, and the final hand design was only slightly different from the Model W.

The scope of this work was intentionally limited to symmetric planar manipulators, as exhaustively exploring the design space of planar hands is more tractable than that of spatial hands. Despite this limitation, we also investigated all asymmetric planar designs comprised of pairwise combinations of the designs shown in this work. In general, we found that symmetric hands outperformed asymmetric hands in nearly every case, and the top performing symmetric hands far outperformed the top asymmetric hands. Due to the extremely large number of resultant designs, the difficulty of visually capturing their performance in a figure, and their relatively poor performance overall, they were excluded from this work. However, the future design of a spatial hand will undoubtedly benefit from some of the lessons learned from this study, regardless of its simplifying assumptions.

#### 5.3.2 Experimentation

One limitation of this work is the grasping strength of the hand. Because the hand was designed with inexpensive hobby motors, and because the finger surfaces were left as unfinished

low-friction ABS plastic to facilitate easy rolling and sliding, the grasp strength of the hand was quite low. A second iteration of this hand could benefit from both stronger motors and from actuated friction pads that can retract into the fingers on command such as in [78].

Despite the rather low grasping force, this hand was highly capable of manipulating a wide range of objects. Specifically, the hand was easily able to perform continuous object rotation, pinch to power transitions, simultaneous manipulation of multiple objects at once, manipulation in the presence of external disturbances, manipulation of soft objects, and object position and orientation servoing within the hand. The hand demonstrated very robust manipulation in the presence of continually changing and unpredictable contact conditions (e.g. multiple sliding and rolling contacts between the object and the hand that were constantly broken and reestablished).

Often, the same open loop strategies worked for multiple objects of different sizes, due to the low torque threshold set for the motors. In a sense, the torque limits produced a compliant finger behavior—a large enough force applied to the object by one finger could backdrive the opposing finger, depending on its relative mechanical advantage. Most notably, the hand almost never lost an object from its grasp. The inclusion of caging in the metric had a huge impact on the range of experiments we were able to attempt, as we rarely had to deal with object ejection, and could therefore be more adventurous in the manipulation strategies attempted through control, regularly finding success in physical demos during the first few tries.

# 6 ENERGY MODEL DESIGN FOR SPATIAL

## CAGING MANIPULATION

When all surfaces of a robotic hand are used for manipulation, rather than just the

fingertips, its dexterity can be markedly increased [12]. It is this principle upon which our work is based. This observation has been made by other researchers within the field of robotic manipulation in the past and can be easily understood simply by watching humans manipulate. While many traditional models of manipulation are built around an assumption of fixed fingertip contact between the hand and the object [58, 59], this work allows contacts to be established and broken freely along all inner surfaces of the hand. This is possible because of the energy-based forward motion model at the heart of this work which does not require precise modeling of contact mode, friction, or contact forces. This is in contrast to previous work in the literature [60, 61, 62], which shows that accurate prediction of contact motion depends on precisely measuring contact force magnitude and direction, as well as coefficient of friction which is known to change over time [63], and local surface geometry [64, 65, 66]. This is a tradeoff—namely the model is based on many simplifying assumptions—but as shown in this work and in previous work [7], these assumptions lead to favorable results in many common manipulation scenarios.

This work shows how the potential energy-based motion model from [7] can be formulated for spatial manipulation in three dimensions, and how it can be used to design spatial manipulators. The work begins in simulation and culminates in the development and testing of a novel spatial manipulator called the Model B. This hand has four fingers, each consisting of fully actuated two link serial chains; two opposing fingers on prismatic bases and two on a rotary base for abduction about an axis orthogonal to the palm. The hand was designed to perform manipulation while maintaining a cage on the object [1, 2, 3], allowing contact constraints to be relaxed and helping to enable more adventurous manipulation primitives without increasing the risk of object ejection.

Many robotic hands have been designed specifically to achieve some sort of manipulation or grasping task. The utility of certain design features, such as underactuation or compliance, has long been embraced for graspers, as it enables easy and robust open-loop grasping of novel object geometries [55]. Researchers have often taken these simple underactuated hands a step further than grasping, showing that they can be used for simple open-loop manipulation tasks as well [13,4]. It has been shown that with subtle design changes, these simple underactuated hands can have their manipulation capability greatly increased for certain tasks, such as planar rotation of an object [68, 32]. Other hands have been designed to achieve large amounts of object translation [20] within the hand, or even both rotation and translation [69, 71].

There are some hands that have been designed specifically for planar whole-hand manipulation, and even fewer for whole-hand spatial manipulation. Those that have been designed for whole-hand manipulation often have tens of degrees of freedom (DOF) and need complex controllers to perform manipulation primitives [80, 81]. This illustrates the classic tradeoff between added DOF / control authority, and increased controller complexity. This work shows how, with careful design, useful dexterity can be achieved without an excessive number of actuators, without necessitating overly complex control or expensive sensors. In the literature there are many examples of controllers ranging in complexity from pure kinematics [29, 41], to more detailed controllers with contact modelling of rolling and sliding [72], to those that include full blown dynamics modelling [8].

Those controllers that include contact modelling typically implement force-closure conditions to ensure that an object is not dropped during manipulation. In this work however, the Model B was designed to maintain a loose cage on an object while manipulating. Future more optimized hand designs could very well utilize existing models of spatial caging [16] during the design process to guarantee a cage over a wide range of object shapes and sizes.

This work shows how the potential energy-based motion model from [7] can be formulated for spatial manipulation in three dimensions, and how it can be used to design spatial manipulators. The work begins in simulation and culminates in the development and testing of a novel spatial manipulator called the Model B. This hand has four fingers, each consisting of fully actuated two link serial chains; two opposing fingers on prismatic bases and two on a rotary base for abduction about an axis orthogonal to the palm. The hand was designed to perform manipulation while maintaining a cage on the object [16, 17, 18], allowing contact constraints to be relaxed and helping to enable more adventurous manipulation primitives without increasing the risk of object ejection.

#### 6.1 Energy-based Forward Motion Model

The manipulation model described in this work extends previous work done on the design of a dexterous hand for planar caging manipulation to three dimensions. This theory is based on the idea that with enough mechanical work, a position controlled motor can be back driven past its set point, gaining potential energy during the process. It was originally inspired by the work of energy-based analysis of grasping in [20, 74]. All nomenclature used in this section is defined in Table 1. The gained potential energy of this motor gained during a displacement is

$$U = \tau_m \big(\theta_d - \theta_{sp}\big) \tag{1}$$

where  $\tau_m$  is the torque supplied by the motor,  $\theta_{sp}$  is the commanded set point of the motor, and  $\theta_d$  is the new position of the shaft after being displaced. A displacement can occur due to an external disturbance that overcomes the torque of the motor. Such a system may include a transmission, which can be represented by *K*, which simply relates the amount of joint displacement to the actual displacement felt by the motor. In a system with multiple actuators, the overall potential energy gained by their combined displacements can be represented by

$$U = \sum_{i=1}^{n} \tau_i K_i (\alpha_i - \theta_i)$$
<sup>(2)</sup>

where there are n actuators in the system. As an example, a two-link finger can be displaced at the fingertip by contacting an immovable object, displacing the motors at each of its joints. In this scenario, we can then calculate the potential energy gained by the motors during this displacement. This is illustrated in Figure 6.1.

## 6.1.1 System Kinematics

Using forward kinematics, the displacement of a motor or a joint can easily be mapped to a corresponding Cartesian displacement of a robot link. Specifically, we can use transformation matrices constructed from Denavit-Hartenberg parameters to relate these parameters. The Cartesian position of a frame affixed to joint k of a serial chain linkage with n links in a fixed global frame can be extracted from the transformation matrix  ${}^{0}T_{k}$  as  $u_{k}$ , where

$${}^{0}T_{k} = \begin{bmatrix} R_{k} & {}^{0}\boldsymbol{u}_{k} \\ 0 & 1 \end{bmatrix} = \prod_{i=1}^{k} {}^{i-1}T_{i}(\boldsymbol{\theta}_{i}), \quad k \leq n \quad {}^{0}\boldsymbol{u}_{k} \in \mathbb{R}^{N} \quad (3)$$

such that  ${}^{i-1}T_i(\theta_i)$  is the homogeneous transformation matrix that transforms points from link i - 1 to link i,  ${}^{0}u_k$  is the joint position vector for the *k*th joint, and  $R_k$  is the rotation matrix associated with the *k*th joint. In this work we assume that all standard Denavit-Hartenberg conventions apply, and also that a joint cannot hyperextend past the previous link. In other words,



Fig. 6.1. A four-fingered hand in simulation with a cube. The pink links show the commanded finger positions, the blue links show the displaced finger positions due to the geometry of the cube. The potential energy of the system is due to the difference in the commanded and displaced joint positions. A) isometric view; B) side view

the robotic fingers described in this work can curl inwards on one extreme, and form a straight serial chain on the other. For each finger, let the Cartesian positions of each joint be

contained in the set  $P_{\theta_{sp}}$  at the commanded actuation input  $\theta_{sp}$  such that

$$P_{\theta_{sp}} = \left\{ {^{k-1}u_{\theta_{sp},k} \middle| k = 1, \dots, n} \right\}, {^{k-1}u_k} \in \mathbb{R}^3$$
(4)

81

Let there be an object to be manipulated described by a set S of m boundary points with respect to a joint-affixed frame i such that

$${}^{i}S_{q} = \{ {}^{i}s_{q,ij} | i = 1 \dots n, j = 1, \dots, m \}, {}^{i}s_{j} \in \mathbb{R}^{3}$$
(5)

where  ${}^{i}S_{j} \in \mathbb{R}^{3}$  for spatial objects. The challenge of this work is to determine the displacement at the joint or motor level caused by the hypothetical forcible placement of an immovable object, such that it displaces the hand. In other words, given a known object pose and hand configuration, how is the hand displaced assuming the object cannot be moved? And furthermore, what is the associated potential energy gained by the hand due to this displacement? These questions are answered by formulating the problem as an optimization program wherein the overall system energy is minimized subject to kinematic and non-penetration constraints that keep the fingers from passing through the object. This problem is challenging because we do not know exactly where the fingers will contact the object, and contact anywhere along the surfaces of the fingers or palm is feasible—unlike more traditional models of robotic manipulation which assume object contact only occurs at the fingertips.

#### 6.1.2 Energy Minimization

Given an external displacement, the hand will reconfigure to its lowest energy configuration. Thus, we formulate an optimization program to minimize the system's total potential energy subject to constraints. The optimization program results in the scalar minimal system energy value as well as the associated minimal energy configurations of all joints of hand. It is formulated as follows

$$U_{\theta_{sp},q}^{*} = \min_{\theta_{d}} \sum_{i=1}^{n} \tau_{i} K_{i} (\theta_{d_{i}} - \theta_{sp_{i}})$$
(6)  
s.t.  $0 \le \theta_{d_{i}} \le \theta_{sp} \quad \forall i = 1 \dots n$   
 $f(x) \le 0$ 

where  $U_{\theta_{sp},q}^*$  is the minimum system energy given commanded actuation input  $\theta_{sp}$  and object pose q. The optimal solution is a vector  $\theta_d^*$  consisting of the displaced joint positions. The constraint  $f(x) \leq 0$  in this optimization program represents a generalized non-penetration or collision constraint that prevents the fingers from penetrating the object, and  $0 \leq \theta_{d_i} \leq \theta_{sp}$ prevents the joints from exceeding their travel limits (due to mechanical hard-stops at each joint). The collision constraint can be implemented in any number of ways, but it is advantageous to find a method that is computationally efficient since it will need to be calculated many times during minimization. The resulting joint positions must also be less than the commanded set points, since the object cannot pull the finger past where it is actuated. For a hand with multiple fingers, we can simply solve the optimization program once for each finger and sum the resultant energies to obtain the overall potential energy of the hand at each configuration of the system.

#### 6.1.3 Energy Fields

Solving the energy minimization for all feasible poses of the object where manipulation is possible yields the energy field for a given system configuration and actuation input, represented by

$$U_{\theta_{sp}} = \left\{ U_{\theta_{sp},q}^* \middle| q = 1, \dots, Q \right\}$$
(7)

where  $U_{\theta_{sp}}$  is a scalar field that shows how an object will move when manipulated. Namely, it enables visualization of the system's potential energy contours, including the workspace region containing the lowest energy, which is where the object will be most likely to settle once the given actuation has been applied to the hand.

## 6.1.4 Gradient of Energy Map

The energy field gradients result in vector fields, lending even more intuition about the motion of an object given a system configuration and actuation input. Specifically, the vector field consists of net wrench vectors that will be applied to the object under the given assumptions. This vector field can be written as

$$\gamma_{\theta_{sp}} = -\nabla_{x, yz, \beta, \phi, \zeta} \mathbf{U}_{\theta_{sp}} \tag{8}$$

for each scalar field  $\mathbf{U}_{\theta_{sp}}$  with actuation input  $\theta_{sp}$ . The vectors comprising the field  $\gamma_{\theta_{sp}}$ are concretely wrench vectors  $w_q \in \mathbb{R}^6$  are of the form  $w_q = [f_{xq}, f_{yq}, f_{zq}, \tau_{xq}, \tau_{yq}, \tau_{zq}]$  and each could be potentially realized by the object—potentially because the model does not take friction into account, ideal object and hand geometry are assumed, and actuation is assumed to be ideal. Thus, the actuation that produces these wrenches is a necessary, rather than a sufficient condition for the physical existence of these wrenches. The set of all vector fields  $\Gamma = \{\gamma_1, ..., \gamma_{\Theta}\}$  calculated over the set  $\Theta$  of all possible actuator inputs is useful for evaluating the overall manipulation capabilities of a hand. For a given system pose, the set of all possible vectors corresponding to all possible actuation inputs can be written as  $\mathbf{W}_q = \{w_{q,1}, ..., w_{q,\Theta}\}$  where the span of these vectors represent all possible wrenches that could be applied to the object in its current configuration.

## 6.1.5 Manipulation Metric

The convex hull of the set  $\mathbf{W}_q$  of all possible vectors that can be applied to the object in a given configuration is  $Conv(\mathbf{W}_q)$ . The radius of the largest ball that can be inscribed inside this hull represents the largest wrench that can be imparted to the object in any direction. This is a useful metric for judging the hand's manipulability. Considering the average, minimum, or maximum radius over the hand's whole workspace tells us useful information about its overall manipulability. This can be used to compare one hand design to another. In this work, we consider the average radius of the largest ball over the hand's entire workspace,  $M_{avg}$ . The algorithm used to perform this calculation is detailed in our previous work [7] and in the literature [75, 35]. The larger the inscribed ball, the larger the wrench that can be exerted on an object in any direction. The radius of this ball (or n-sphere) is calculated by first finding the polytope representation of  $Conv(\mathbf{W}_q)$ , which is defined by the intersection of hyperplanes  $\langle h_i, w \rangle \leq b_i$ . Given the hyperplane representation, the maximum radius can be computed with the quadratic program

$$w_{\mathbf{q}}^{*} = \max_{w} ||w||^{2}$$
(9)  
s.t.  $\langle h_{1}, w \rangle \leq b_{1}$   
 $\langle h_{2}, w \rangle \leq b_{2}$   
 $\vdots$   
 $\langle h_{n}, w \rangle \leq b_{n}.$ 

In this work, the ball lives in a six dimensional space (the spatial wrench space). The score for a hand  $M_{avg}$  is the average radius over entire workspace, averaged over all objects.

## 6.2 Simulation of Robotic Hands

Eleven commercially available, open-source, or novel robotic hand topologies were simulated in this work and their manipulation capabilities were quantified based on the metric described in the previous section. The goal of this work was not to find an optimal design for spatial manipulation in a strict sense, but simply to explore the space of existing designs and hypothesize how a more dexterous hand might look and function. The simulation of these hands demonstrates that the energy-based motion model can be used with hands consisting of many actuators, such as the Allegro hand (16 actuators). The theoretical manipulability of each hand was simulated on the Yale High Performance Computing resource, taking anywhere from less than a minute to several days depending on the complexity of the hand.

To begin, five existing hand designs were simulated, as shown in Figure 6.2. These include commercially available hands such as the Allegro hand, as well as open-source hands such as the Yale OpenHand Model T42, Model Q, and Model O [57, 79, 28, 53]. We found that simulating these hands, especially the OpenHand models, was very valuable in establishing intuition about



Fig. 6.2. Eleven hands were simulated to manipulate cubes in this work throughout their workspace. The kinematic topologies for are shown for the following simulated hands: A) underactuated and fully actuated T42; B) Allegro Hand; C) underactuated and fully actuated Model O; D) underactuated and fully actuated Model Q; E) H1; F) H2; G) Model B; H) simulation world frame; I) simulated cubes; J) workspace grid (it is centered with the palm and raised one half cube side length along the z-axis)

the overall performance in known terms, as there are many examples of these hands manipulating various objects within the literature. Six novel hand designs were also simulated. These include fully actuated versions of the Yale OpenHand Model T42, Model Q, and Model O, as well as completely new hand topologies. In this work we refer to the new topologies as the H1, the H2, and the Model B. The Denavit-Hartenberg parameters used to simulate all of these hands are shown in Table 6.1. In the case of underactuated hands, the joint coupling is also noted.

## 6.3 Simulation Results

The average manipulability was calculated for each hand according to them metric in equation 9 and the results are shown in Figure 6.3. The results show that in general, hands with more actuators perform better—but not exclusively so. Some hands with many motors, such as the Allegro hand, do not perform very well, likely do to the kinematic redundancy of its fingers. After all, the Allegro Hand was likely designed for fingertip manipulation, rather than whole-hand manipulation as is being assessed in this work. It seems that in order to achieve higher dexterity throughout the workspace in a whole-hand sense, it is advantageous to use sets of opposing fingers.

Several hands including the underactuated and fully actuated versions of the T42 and the underactuated version of the Model O had virtually negligible spatial manipulation capability. This matches nicely with intuition, as the T42 topologies can clearly only manipulate in the plane of the fingers, and have no control authority orthogonal to that plane. This means that in theory, manipulation capability should be zero according to the metric. The underactuated Model O suffers from a similar problem, although it does have the ability to theoretically manipulate in more directions. In its case, the low manipulability instead comes from its underactuation, rather than its kinematic topology.

## 6.4 The Model B Hand

A physical hand was designed based on the simulated design parameters and it is shown as a CAD rendering in Figure 6.4. The hand has eleven inexpensive smart servos, specifically Dynamixel XL-320's. It has an 80mm diameter palm, two opposing prismatic fingers with 72mm proximal links and 46mm distal links. It also has a pair of coupled abduction fingers with 30 degrees of rotation about the center axis of the palm. Each abduction finger has a 57mm proximal link and a 46mm distal link. All parts, including gears and rack, were 3D printed out of ABS using
a Stratasys uPrint. Revolute joints were realized using ball bearings and shoulder bolts, and the prismatic joints were supported using off the shelf carriages and rails, and actuated by rack and pinion.

Bench top experiments were performed to assess the hand's ability to manipulate a variety of objects in a controlled environment. Open loop manipulation primitives were manually determined and hard coded such that manipulation motions could be chained together. These manipulation primitives include left-right shift motions, power to pinch motions, pinch to power transitions, roll motions, and yaw motions—both in clockwise and counter clockwise directions, both right side up and upside down.



Fig. 6.3. Top Panel: the hands simulated in this work are shown and sorted by the number of actuators. An 'F' in a hand's name indicates that it is the fully actuated version of an underactuated hand from Yale OpenHand. Bottom Panel: The same hands are now sorted by their performance based on the manipulation metric  $M_{avg}$  described in section II.F.

#### 6.5 Experimental Results

The hand's robust manipulation ability was demonstrated by performing repeated manipulation primitives both in the bench top setting and on a 7 degrees of freedom (DOF) Barrett Whole Arm Manipulator (WAM) robotic arm. These demonstrations were performed using the open loop motion primitives described in the previous section. A variety of objects were



Fig. 6.4. A rendering of the Model B hand. It is comprised of four fingers—two pairs of identical opposing fingers. One set is prismatic, the other rotates about the axis of the palm. Each finger has a proximal and distal link. The distal links of each finger interdigitate with the opposing finger.

manipulated including some from the Yale-CMU-Berkeley (YCB) Object Set [77], as well as painted wooden cubes and a soft knitted cube.

First, it was demonstrated that the hand could successfully perform the roll, yaw, and leftright shift primitives for the wooden cubes, knitted cube, foam cube, and rubber duck continuously in both directions with gravity pointing downwards into the palm. These motions were performed on loop and video of the task was recorded. Next, the hand was set up to perform a power to pinch manipulative motion of the red ball repeatedly, and video was taken of the task (Figure 6.6). Next, these tasks were repeated with the painted wooden cube with the hand flipped upside down, so that gravity pointed away from the palm. An additional motion was programmed into the controller that would pick up the cube from a surface below the hand, essentially completing a pinch to power primitive against gravity (Figure 6.5).

After performing bench top tasks, the hand was fitted to the WAM arm and made to grasp a wide array of objects. These objects included a painted wooden block, a plastic orange, stackable



Fig. 6.5. The Model B can manipulate a wooden cube against gravity. A) The hand performs a "yaw" motion with the cube; B) The hand performs a "roll" motion with the cube; C) The hand performs a pinch to power transition against gravity, drawing the cube up from the support surface into a power grasp.

plastic cups, a rubber duck, a 3D printed Stanford bunny, a plastic toy car, a golf ball, and a dice. These tasks were performed to show the range of object sizes that can easily be accommodated by the hand. Next, the hand was commanded to perform the roll task on the knitted cube, as the arm continuously moved the hand through different configurations in space (Figure 6.8). This task demonstrates that the manipulation of some objects is not dependent on a fixed gravity vector. Next, the hand was made to continually manipulate the painted wooden cube using the yaw and roll motions while the hand's configuration was continuously moved through space. This was possible because of the caging grasps that were maintained during all manipulation primitives, allowing the hand to move an object but not drop it. Finally, the hand was programmed to grasp and manipulate four painted blocks, changing their upward facing letters from G-R-A-B to Y-A-

		Finger	Joint	DH Parameters				
				Type	а	α	d	θ
Hand	T42	1,2	1	R	0.63	0	0	0
			2	R	0.37	0	0	0
	Model O	1,2	1	R	0.403	$\pi/2$	0	0
			2	R	0.63	0	0	0
			3	R	0.37	0	0	0
		3	1	R	0.63	0	0	0
			2	R	0.37	0	0	0
	Q	1,2	1	R	0.6	0	0	0
			2	R	0.4	0	0	0
	Model	3,4	1	R	0.25	$\pi/2$	0	0
			2	R	0.6	0	0	0
			3	R	0.4	0	0	0
	H1	1,2	1	R	0	$\pi/2$	0	0
			2	Р	0	$\pi/2$	0.5	$-\pi/2$
			3	R	0.5	0	0	0
		3	1	Р	0	$\pi/2$	0.5	0
			2	R	0.5	0	0	0
	H2	1-4	1	Р	0	$\pi/2$	0.5	0
			2	R	0.5	0	0	0
	Model B	1,2	1	Р	0	$\pi/2$	0.33	0
			2	R	0.33	0	0	0
			3	R	0.33	0	0	0
		3,4	1	R	0.25	$\pi/2$	0	0
			2	R	0.6	0	0	0
			3	R	0.4	0	0	0
	Allegro	1-3	1	R	0	$\pi/2$	0	0
			2	R	0.54	0	0	0
			3	R	0.384	0	0	0
			4	R	0.387	0	0	0
		4	1	R	0.621	0	0	0
			2	R	0	$-\pi/2$	0	0
			3	R	0.514	0	0	0
			4	R	0.387	0	0	0

Table 2. Parameters for simulated hands

L-E, chaining together manipulation primitives and grasps to accomplish a real world task. First, this task was performed with gravity facing downwards into the palm (the hand facing upwards) (Figure 6.7). Last, this task was completed again with the hand upside down, enabling the task to be completed much faster without a large motion of the arm to reconfigure the hand (Figure 6.9). Video was recorded of all grasps and WAM manipulation demos and can be seen in the supplementary media attachment.

### 6.6 Conclusion

This paper presents the first ever formulation of the potential energy based forward motion model to three dimensions. We demonstrate how that can be used to compare the theoretical spatial manipulation performance of hand designs, and use this comparison to design a new hand, the Model B. The physical open loop manipulation capabilities of this hand were demonstrated and are shown to be capable of robustly manipulating a variety of objects.

There are many strong assumptions that went into this work that are worthy of discussion. Namely, the potential energy based forward motion model does not take friction into account. To date, we have not investigated the role that friction plays in the utility of this model. Also, at best we have only created a discrete representation of the actuator capabilities of each hand, meaning that a hand's true continuous performance will never be fully captured using the methods in this paper. A large reason for this is that the more densely actuation input is sampled, the more



Fig. 6.6. The Model B can manipulate objects with gravity into the palm. A) The hand performs a "yaw" motion with the wooden cube; B) The hand performs a left-right shift with a wooden cube; C) The hand performs a "roll" motion with the knitted cube; D) The hand performs a power to pinch transition against gravity with a ball.



Fig. 6.7. The Model B reoriented four painted cubes from G-R-A-B to Y-A-L-E, transitioning the hand to a vertical configuration before performing any manipulation primitives.

computationally prohibitive the problem becomes. Despite these assumptions, we believe that in many cases the energy model is a useful tool and does a good job of approximating the motion of an object (see [11] for data on this). To that end, we believe that it would be particularly useful as a motion model in a real time closed loop controller for manipulation, though that must be saved for future work. While our previous work [11] designed hands specifically for good manipulation while caging, this work did not quantify the caging ability of hands. The reason for this is that it is much more challenging and computationally expensive to implement a metric for spatial caging than the planar caging technique used in our previous work. Rather than try to quantify a hand's caging abilities, we instead chose to use our intuition to design a hand that could reasonably cage



Fig. 6.8. The Model B continuously reoriented the knitted cube while the WAM moved the hand through space, constantly changing the hand's orientation with respect to gravity.

objects within a certain size range. Indeed, looking at the results of our experimental work, it is clear that the Model B is successful at caging objects of a certain size, as it can perform manipulation primitives while changing the configuration of the hand with respect to the gravity vector without object ejection.



Fig. 6.9. The Model B performed manipulation tasks with gravity pointing away from the palm. A) The hand begins to reorient four painted wooden cubes from Y-A-L-E to G-R-A-B; B) The hand performs a sequence of pre-determined open-loop manipulation primitives on the first cube, transitioning its front-facing side from the letter Y to the letter G; C) The hand completes the task of reorienting all cubes; D) The hand performs a "yaw" motion to the cube while moving through space; E) The hand performs a "roll" motion to the cube while moving through space; F) the hand performs a "yaw" motion

# 7 CONCLUSION

#### 7.1 Summary

In this work I explored a new way to think about robotic manipulation. Specifically, we considered that caging, rather than force or form closure, could be used to guard against object ejection, enabling higher risk manipulation strategies involving rolling, sliding, and constantly shifting contacts. I showed how manipulation of this style can be predicted through a potential

energy view of the hand-object system. I demonstrate how the caging energy can be used for control and design of fully actuated and underactuated robotic hands, and how it can be formulated as an optimization program for both planar and spatial systems. I presented two novel robotic hands, the Model W and the Model B, capable of dexterous planar and spatial manipulation respectively. The performance of these hands was validated experimentally in a number of ways, lending credibility to the utility of the energy model as a design tool.

#### 7.2 Lessons Learned and Future Work

Though this work makes progress towards realizing a type of highly dexterous manipulation wherein contact constraints can be more relaxed and fluid, there is still an enormous amount of work to be done to truly bring this theory to its full potential. Specifically, formulating the optimization program in a way that enables real time control of a hand-object system would prove the model's utility in an irrefutable way. Alas, as I am only a single grad student, with a specialty in mechanical design, this is slightly outside the scope of my work. I believe that someone with more expert optimization and programming skills would be very well suited to accomplishing this. I do believe that it can be done, but it was simply more than I had time to do. Next, proving the convexity (or non-convexity) of the energy model for certain general cases would be another helpful direction for this work. Thanks to some unpublished work I did with Dr. Christopher Harshaw, we nearly proved convexity of the energy model from Chapter 5 for certain planar handobject systems—but again, due to scope and time constraints, this was never complete enough to be published. Proving complexity would enable the use of special computational tricks to speed up the energy calculation, which may be necessary for its use in a real-time controller. The convexity of the model is highly sensitive to the formulation of the non-penetration constraint, so potentially a clever formulation of this constraint could easily be convex, but I have not found one.

Further work that would improve the adoption of the energy model in the wider community is perhaps less glamorous but just as important as the real-time controller. Namely, characterizing how the accuracy of the energy model breaks down with increasing friction between the hand and the object, as well as characterizing the model's accuracy in predicting net wrench applied to an object would both be valuable investigations. Thanks to Yale undergraduates Laszlo Kopits and Zubin Kremer Guha, both of these studies have been initiated, but neither are complete as of the writing of this document. Understanding these characteristics of the energy model would help researchers determine when it can be applied to their own systems, or when it instead makes sense to use a different approach.

## REFERENCES

- S. Makita, Y. Maeda, 3D multifingered caging: Basic formulation and planning. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2697-2702.
- R. Diankov, S. S. Srinivasa, D. Ferguson, J. Kuffner, Manipulation planning with caging grasps. In Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots, pp. 285-292. IEEE, 2008.
- A. Rodriguez, M. T. Mason, S. Ferry, From caging to grasping. The International Journal of Robotics Research 31, no. 7 (2012): 886-900.
- 4. R. R. Ma, W. G. Bircher, A. M. Dollar, Toward robust, whole-hand caging manipulation with underactuated hands. In 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 1336–1342.
- W. G. Bircher, A. S. Morgan, K. Hang, and A. M. Dollar, Energy gradient-based graphs for planning within-hand caging manipulation. In 2019 International Conference on Robotics and Automation (ICRA).IEEE, 2019, pp. 2462–2467.
- W. G. Bircher, A. M. Dollar, Design principles and optimization of a planar underactuated hand for caging grasps. In 2019 International Conference on Robotics and Automation (ICRA), pp. 1608-1613. IEEE, 2019.
- 7. Bircher, Walter G., Andrew S. Morgan, and Aaron M. Dollar. "Complex manipulation with a simple robotic hand through contact breaking and caging." Science Robotics 6.54 (2021).
- 8. Z. Li, P. Hsu, and S. Sastry, "Grasping and coordinated manipulation by a multifingered robot hand," *Int. J. Rob. Res.*, vol. 8, no. 4, pp. 33–50, 1989.
- 9. M. T. Mason and J. K. Salisbury, *Robot hands and the mechanics of manipulation*. 1985.
- 10. M. Grebenstein, M. Chalon, W. Friedl, S. Haddadin, T. Wimbock, G. Hirzinger, and R.

Siegwart, "The hand of the DLR Hand Arm System: Designed for interaction," *Int. J. Rob. Res.*, vol. 31, no. 13, pp. 1531–1555, Nov. 2012.

- T. Senoo, Y. Yamakawa, S. Mizusawa, A. Namiki, M. Ishikawa, and M. Shimojo,
   "Skillful Manipulation Based on High-speed Sensory-Motor Fusion," in *International Conference on Robotics and Automation*, 2009, pp. 1611–1612.
- G. Vassura and A. Bicchi, "Whole-Hand Manipulation: Design of an Articulated Hand Exploiting All Its Parts to Increase Dexterity," in *Robots and Biological Systems: Towards a New Bionics?*, 1993, pp. 165–177.
- L. U. Odhner, R. R. Ma, and A. M. Dollar, "Open-loop precision grasping with underactuated hands inspired by a human manipulation strategy," *IEEE Trans. Robot.*, vol. 10, no. 3, pp. 625–33, 2013.
- 14. M. N. Ahmadabadi and E. Nakano, "A "constrain and move" approach to distributed object manipulation," *IEEE Trans. Robot. Autom.*, vol. 17, no. 2, pp. 157–172, 2001.
- W. Wan, R. Fukui, M. Shimosaka, T. Sato, and Y. Kuniyoshi, "Grasping by caging: A promising tool to deal with uncertainty," in *International Conference on Robotics and Automation*, 2012, pp. 5142–5149.
- S. Makita and Y. Maeda, "3D multifingered caging: Basic formulation and planning," in International Conference on Intelligent Robots and Systems, 2008, pp. 2697–2702.
- R. Diankov, S. S. Srinivasa, D. Ferguson, and J. Kuffner, "Manipulation planning with caging grasps," in *IEEE-RAS International Conference on Humanoid Robots*, 2008, pp. 285–292.
- A. Rodriguez, M. T. Mason, and S. Ferry, "From caging to grasping," *Int. J. Rob. Res.*, vol. 31, no. 7, pp. 886–900, Apr. 2012.

- A. Sudsang, J. Ponce, and N. Srinivasa, "Algorithms for Constructing Immobilizing Fixtures and Grasps of Three-Dimensional Objects," no. 2.
- 20. G. A. Kragten and J. L. Herder, "The ability of underactuated hands to grasp and hold objects," *Mech. Mach. Theory*, vol. 45, no. 3, pp. 408–425, 2010.
- A. Rodriguez, M. T. Mason, and S. S. Srinivasa, "Manipulation Capabilities with Simple Hands," *Exp. Robot.*, pp. 1–15, 2014.
- 22. A. Blake, "Caging 2D Bodies by 1-Parameter Two-Fingered Gripping Systems," in *IEEE International Conference on Robotics and Automation*, 1996, no. April, pp. 1458–1464.
- J. Mahler, F. T. Pokorny, Z. Mccarthy, A. F. Van Der Stappen, and K. Goldberg,
  "Energy-Bounded Caging : Formal Definition and 2D Energy Lower Bound Algorithm
  Based on Weighted Alpha Shapes," in *International Conference on Ro*, 2016.
- T. Yamada, T. Koishikura, Y. Mizuno, N. Mimura, and Y. Funahashi, "Stability Analysis of 3D Grasps by A Multifingered Hand," in *International Conference on Robotics and Automation*, 2001, pp. 2466–2473.
- R. R. Ma, L. U. Odhner, and A. M. Dollar, "A Modular, Open-source 3D Printed Underactuated Hand," in *International Conference on Robotics and Automation*, 2013, pp. 2737 – 2743.
- S. Garrido-Jurado, R. Munoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marin-Jimenez,
   "Automatic generation and detection of highly reliable fiducial markers under occlusion,"
   *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, 2014.
- D. M. Aukes, B. Heyneman, J. Ulmen, H. Stuart, M. R. Cutkosky, S. Kim, P. Garcia, and A. Edsinger, "Design and testing of a selectively compliant underactuated hand," *Int. J. Rob. Res.*, Feb. 2014.

- L. U. Odhner, R. R. Ma, and A. M. Dollar, "Exploring Dexterous Manipulation Workspaces with the iHY Hand," *J. Robot. Soc. Japan*, vol. 32, no. 4, pp. 318–322, 2014.
- 29. B. Sundaralingam and T. Hermans, "Relaxed-rigidity constraints: In-grasp manipulation using purely kinematic trajectory optimization," in Robotics: Science and Systems, 2017.
- M. Li, K. Hang, D. Kragic, and A. Billard, "Dexterous grasping under shape uncertainty," Robotics and Autonomous Systems, vol. 75, pp. 352 – 364, 2016.
- 31. J. Trinkle and R. Paul, "Planning for dexterous manipulation with sliding contacts," The International Journal of Robotics Research, vol. 9, no. 3, pp. 24–48, 1990.
- 32. Bircher, Walter G., Dollar, Aarom M., and Rojas Nicolas. "A two-fingered robot gripper with large object reorientation range." Robotics and Automation (ICRA), 2017 IEEE International Conference on. IEEE, 2017.
- 33. K. Tahara, S. Arimoto, and M. Yoshida, "Dynamic object manipulation using a virtual frame by a triple soft-fingered robotic hand," in IEEE International Conference on Robotics and Automation (ICRA), 2010.
- 34. Y. Bekiroglu, J. Laaksonen, J. A. Jorgensen, V. Kyrki, and D. Kragic, "Assessing grasp stability based on learning and haptic data," IEEE Transactions on Robotics, vol. 27, no. 3, pp. 616–629, 2011.
- 35. K. Hang, M. Li, J. A. Stork, Y. Bekiroglu, F. T. Pokorny, A. Billard, and D. Kragic,
  "Hierarchical fingertip space: A unified framework for grasp planning and in-hand grasp adaptation," IEEE Transactions on Robotics, vol. 32, no. 4, pp. 960–972, 2016.
- N. C. Dafle, A. Rodriguez, R. Paolini, B. Tang, S. S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrigge, "Extrinsic dexterity: In-hand

manipulation with external forces," in IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 1578–1585.

- N. C. Dafle, R. Holladay, and A. Rodriguez, "In-hand manipulation via motion cones," in Proceedings of Robotics: Science and Systems, June 2018.
- S. Cruciani, C. Smith, D. Kragic, and K. Hang, "Dexterous manipulation graphs," in IEEE International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018.
- A. Varava, D. Kragic, and F. T. Pokorny, "Caging grasps of rigid and partially deformable 3-d objects with double fork and neck features," IEEE Transactions on Robotics, vol. 32, no. 6, pp. 1479–1497, 2016.
- J. A. Stork, F. Pokorny, and D. Kragic, "A topology-based object representation for clasping, latching and hooking," in IEEE International Conference on Humanoid Robots (HUMANOIDS), pp. 138-145, 2013.
- Maeda, Y. and Asamura, T., 2016, July. Sensorless in-hand caging manipulation. In International Conference on Intelligent Autonomous Systems (pp. 255-267). Springer, Cham.
- Makapunyo, T., Phoka, T., Pipattanasomporn, P., Niparnan, N., & Sudsang, A. (2013, May). Measurement framework of partial cage quality based on probabilistic motion planning. In Robotics and Automation (ICRA), 2013 IEEE International Conference on (pp. 1574-1579). IEEE.
- Makita, Satoshi, and Kazuyuki Nagata. "Evaluation of finger configuration for partial caging." Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE, 2015.

- 44. Besicovitch, Abram Samoilovitch. "A net to hold a sphere." The Mathematical Gazette41.336 (1957): 106-107.
- 45. Kuperberg, Wlodzimierz. "Problems on polytopes and convex sets." DIMACS Workshop on polytopes. 1990.
- 46. O'Rourke, Joseph. "Computational geometry column 18." ACM SIGACT News 24.1 (1993): 20-25.
- 47. Rimon, Elon, and Joel W. Burdick. "Mobility of bodies in contact. I. A 2nd-order mobility index for multiple-finger grasps." IEEE transactions on Robotics and Automation 14.5 (1998): 696-708.
- 48. Rimon, Elon, and Andrew Blake. "Caging planar bodies by one-parameter two-fingered gripping systems." The International Journal of Robotics Research 18.3 (1999): 299-318.
- 49. Pipattanasomporn, Peam, and Attawith Sudsang. "Two-finger caging of nonconvex polytopes." IEEE Transactions on Robotics 27.2 (2011): 324-333.
- 50. Hirano, Daichi, Kenji Nagaoka, and Kazuya Yoshida. "Design of underactuated hand for caging-based grasping of free-flying object." System Integration (SII), 2013 IEEE/SICE International Symposium on. IEEE, 2013.
- 51. Backus, Spencer B., Lael U. Odhner, and Aaron M. Dollar. "Design of hands for aerial manipulation: Actuator number and routing for grasping and perching." Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on. IEEE, 2014.
- Balasubramanian, Ravi, Joseph T. Belter, and Aaron M. Dollar. "Disturbance response of two-link underactuated serial-link chains." Journal of Mechanisms and Robotics 4.2 (2012): 021013.

- Odhner, Lael U., et al. "A compliant, underactuated hand for robust manipulation." The International Journal of Robotics Research 33.5 (2014): 736-752.
- 54. Spanjer, Stefan AJ, et al. "Underactuated gripper that is able to convert from precision to power grasp by a variable transmission ratio." Advances in Reconfigurable Mechanisms and Robots I. Springer, London, 2012. 669-679.
- Dollar, Aaron M., and Robert D. Howe. "The highly adaptive SDM hand: Design and performance evaluation." The International Journal of Robotics Research 29.5 (2010): 585-597.
- Laliberté, Thierry, Lionel Birglen, and Clement Gosselin. "Underactuation in robotic grasping hands." Machine Intelligence & Robotic Control 4.3 (2002): 1-11.
- 57. Ma, Raymond R., & Dollar, Aaron M. Yale OpenHand Project: Optimizing Open-Source Hand Designs for Ease of Fabrication and Adoption. IEEE Robotics & Automation Magazine, (2017): 24(1), 32-40.
- K. Salisbury, B. Roth, Kinematic and force analysis of articulated mechanical hands. Journal of Mechanisms, Transmissions, and Automation in Design, vol. 105, no. 1, pp. 35– 41, 1983.
- J. Kerr, B. Roth, Analysis of multifingered hands. The International Journal of Robotics Research, vol. 4, no. 4, pp. 3–17, 1986.
- D. L. Brock, Enhancing the dexterity of a robot hand using controlled slip. In Proceedings.
  1988 IEEE International Conference on Robotics and Automation. IEEE, 1988, pp. 249–251.

- J. C. Trinkle, A quasi-static analysis of dextrous manipulation with sliding and rolling contacts. In Proceedings, 1989 International Conference on Robotics and Automation. IEEE, 1989, pp. 788–793.
- R. D. Howe, M. R. Cutkosky, Practical force-motion models for sliding manipulation. The International Journal of Robotics Research 15, no. 6 (1996): 557-572.
- 63. K. Yu, M. Bauza, N. Fazeli, A. Rodriguez, More than a million ways to be pushed. A highfidelity experimental dataset of planar pushing. In 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp. 30-37. IEEE, 2016.
- T. Okada, Computer control of multijointed finger system for precise object-handling.
   IEEE Transactions on Systems, Man, and Cybernetics, vol. 12, no. 3, pp. 289–299, 1982.
- A. Cole, J. Hauser, S. Sastry, Kinematics and control of multifingered hands with rolling contact. In Proceedings. 1988 IEEE International Conference on Robotics and Automation, pp. 228-233. IEEE, 1988.
- D. J. Montana, The kinematics of contact and grasp. The International Journal of Robotics Research 7, no. 3 (1988): 17-32.
- 67. W. Yuan, S. Dong, E. H. Adelson, Gelsight: High-resolution robot tactile sensors for estimating geometry and force. Sensors 17, no. 12 (2017): 2762.
- 68. R.R. Ma, N. Rojas, A.M. Dollar, Spherical hands: Toward underactuated, in-hand manipulation invariant to object size and grasp location. Journal of Mechanisms and Robotics, 8(6).
- A. Kakogawa, H. Nishimura, S. Ma, Underactuated modular finger with pull-in mechanism for a robotic gripper. In 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO) (pp. 556-561). IEEE.

- C. M. McCann, A. M. Dollar, Design of a Stewart platform-inspired dexterous hand for 6 DOF within-hand manipulation. In 2017 IEEE/RSJ International Conference on Intelligent
   Robots and Systems (IROS) (pp. 1158-1163). IEEE.
- V. Tincani, M. G. Catalano, E. Farnioli, M. Garabini, G. Grioli, G. Fantoni, A. Bicchi, Velvet fingers: A dexterous gripper with active surfaces. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1257-1263. IEEE, 2012.
- 72. J. C. Trinkle, R. Ram, A. Farahat, P. F. Stiller, Dexterous manipulation planning and execution of an enveloped slippery workpiece. In 1993 Proceedings IEEE International Conference on Robotics and Automation. IEEE, 1993, pp. 442–448.
- 73. R. R. Ma, W. G. Bircher, A. M. Dollar, Modeling and evaluation of robust whole-hand caging manipulation. IEEE Transactions on Robotics 35, no. 3 (2019): 549-563.
- M. T. Mason, A. Rodriguez, S. S. Srinivasa, A. S. Vazquez, Autonomous manipulation with a general-purpose simple hand. The International Journal of Robotics Research 31, no. 5 (2012): 688-703.
- 75. S. Boyd, L. Vandenberghe, Convex optimization. Cambridge university press.
- 76. G. M. Ziegler, Lectures on polytopes. Vol. 152. Springer Science & Business Media, 2012.
- 77. B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, A. M. Dollar, The YCB object and model set: Towards common benchmarks for manipulation research. In 2015 international conference on advanced robotics (ICAR), pp. 510-517. IEEE, 2015.
- A. J. Spiers, B. Calli, A. M. Dollar, Variable-friction finger surfaces to enable within-hand manipulation via gripping and sliding. IEEE Robotics and Automation Letters 3, no. 4 (2018): 4116-4123.

- 79. Raymond R. Ma and Aaron M. Dollar "An Underactuated Hand for Efficient Finger-Gaiting-Based Dexterous Manipulation," proceedings of the 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2014.
- Andrychowicz, OpenAI: Marcin, et al. "Learning dexterous in-hand manipulation." The International Journal of Robotics Research 39.1 (2020): 3-20.
- 81. ShadowRobot (2005) ShadowRobot Dexterous Hand. https://www.shadowrobot.com/products/dexterous-hand/