
# Design Iteration of Dexterous Compliant Robotic Manipulators

Pragna Mannam

CMU-RI-TR-24-01

Ph.D. Thesis

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

**Thesis Committee:**

| | |
|---|---|
| Dr. Nancy Pollard | Carnegie Mellon University (Co-Chair) |
| Dr. Jean Oh | Carnegie Mellon University (Co-Chair) |
| Dr. Matthew T. Mason | Carnegie Mellon University |
| Dr. Oliver Brock | Technische Universität Berlin |

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

**Keywords:** Mechanism design, soft robots, manipulation, design optimization

# Abstract

The goal of personal robotics is to have robots in homes performing everyday tasks efficiently to improve our quality of life. Towards this end, manipulators are needed which are low cost, safe around humans, and approach human-level dexterity. However, existing off-the-shelf manipulators are expensive both in cost and manufacturing time, difficult to repair, and unsafe to operate with delicate objects due to rigid components. Soft robotic manipulators, on the other hand, show great promise as their compliance allows them to conform to objects, exhibit physical robustness, and execute safe object interactions, while being low cost through the use of rapid prototyping. However, designing soft robot manipulators is challenging due to their high degrees of freedom and the inherent complexity of modeling soft materials in simulation. This makes it difficult to iterate on the soft manipulator design prior to manufacturing. To this end, this thesis explores a spectrum of approaches to quickly iterate on the design and evaluation of soft hands, in a matter of days, to allow rapid turnaround of real robot hand prototypes. This allows the designer to swiftly assess the performance of the design for tasks in the real world, and inform future design improvements using real-world measurable metrics. We explore various design iteration approaches by designing and evaluating two different types of manipulators: a parallel delta manipulator and a tendon-driven anthropomorphic (human-like) hand.

Delta manipulators have high precision and low inertia, which lend themselves perfectly to performing fine-grained manipulations. However, 3D-printing these manipulators with soft materials leads to non-ideal kinematic behavior. First, we explore iterating on the design of 3D-printed parallelogram links, which are a key component of the delta manipulator, using a human-in-the-loop approach towards achieving close to ideal kinematic behavior. Then, we evaluate a two-fingered 6-DoF delta manipulator, consisting of the designed parallelogram links, using teleoperation in real-world tasks. We demonstrate the compliance and dexterity of our gripper through six dexterous manipulation tasks involving small and delicate objects, such as twisting a grape off a stem.

Subsequently, this thesis explores design iteration for anthropomorphic soft hands to achieve contact-rich manipulation that enables a wider variety of tasks. Contrary to first designing the manipulator and then evaluating it in the real world, we create a unified design iteration and evaluation framework for a 3D-printed 16-DoF dexterous anthropomorphic soft hand (DASH). We rapidly design and test five iterations of DASH, in a matter of days, by leveraging 3D-printing for fabrication and utilizing teleoperation for evaluation across 30 real-world manipulation tasks. The changes to each successive iteration of DASH is informed by a human designer after observing previous evaluation results. Our final iteration of DASH solves 19 of the 30 tasks compared to Allegro, a popular rigid manipulator on the market, which can only solve 7 tasks.

Finally, we automate the design iteration process, thereby taking the human out of the loop, in order to explore a larger design space. To achieve this, we simulate the soft hand as a rigid body and use hand evolution methods to transfer control

policies across a large set of generated hand designs. Subsequently, we evaluate the optimized design as a soft tendon-driven hand and show that it solves 23 of the 30 tasks, outperforming our previously iterated hands designs.

This thesis explores techniques for iterative design including rapid prototyping, teleoperation, and simulation to enable designers to tune kinematics and compliance for dexterous tasks through real robot evaluation. We show that substantial improvements can be made between design iterations and over state of the art dexterous robotic hands on dexterity benchmarks. The key takeaway from our work is that achieving robot hand dexterity requires detailed attention to hand kinematics and compliance. We also observe that a rich suite of tasks involving a variety of objects and using non-binary evaluation metrics can better inform the design iteration process. The thesis concludes by presenting several avenues for future work in the design iteration space.

# Acknowledgments

I have met some incredible people during my PhD and have countless people to thank for having come this far. First, I would like to thank my advisors, Jean and Nancy for seeing my potential and supporting me through both the challenges and triumphs. I have known Jean since my days on Girls of Steel, a FIRST Robotics team based out of CMU. And I had the pleasure of working with Nancy even on my Masters thesis committee. I am extremely grateful that we forged this collaboration that resulted in amazing work and a great experience.

For the last ten years, I have been working in robotic manipulation and it is all thanks to Matt. I was lucky enough to be introduced to him through Patti Rote and began working with him in freshman year of my undergraduate program. I was immediately excited by the field and had great discussions with former and current lab members, Xianyi Cheng, Ankit Bhatia, Robbie Paolini, Yifan Hou, Eric Huang, Jiaji Zhou, Nikhil Chavan-Dafle, and more. Special thanks to Matt for being on my thesis committee and for embarking on this incredible journey with me.

Oliver Brock has been a tremendous help in introducing and guiding me through the field of design optimization. I felt less lost in this area through our many discussions, and I am excited about the potential of this research. Thank you for taking the time and being on my committee.

I also want to thank all of my collaborators, without whom this work would not have been possible. Thanks to Oliver Kroemer, Zeynep Temel, Avi Rudich, Kevin Zhang, Dominik Bauer, Kenneth Shaw, Aditya Kannan, Shikhar Bahl, Deepak Pathak, and Xingyu Liu.

The PhD journey was all the more enjoyable from the friends I made and memories I made. Thank you to both of my labs BIG and FRL for their continued support and presence. I also want to thank my friends who are some of the brightest and most amazing individuals, Michelle Zhao, Abena Boadi-Agyemang, Sudharshan Suresh, Abhijat Biswas, Saumya Saxena, Cornelia Bauer, Raunaq Bhirangi, Tess Hellebrekers, Tabitha Lee, Arpit Agarwal, Terri Kent, Nadine Chang, Thomas Weng, Tanmay Shankar, Dhruv Saxena, Arjav Desai, Mike Lee, Ben Eisner, Ami Sawhney, Emily Pruc, Adithya Murali, Xuning Yang, Aditya Dhawale, Brian Okorn, Sudeep Dasari, Jason Zhang, Alex Spitzer, Puneet Puri, Vishal Dugar, Jerry Hsiung, Senthil Purushwalkam, Achal Dave, Rogerio Bonatti, Micah Corah, and many more. Thank you for making the experience feel bigger and brighter than I could have imagined.

Last but not least, I want to thank my family and my partner for always believing in me. My passion for education was inspired by my grandfather, who earned a PhD in Chemistry. My grandmother and he have always been my biggest supporters, instilling in me great confidence. Thank you to my parents for allowing me to soar, knowing that they would always catch me if I fall. I also want to thank my sister, Harshitha, for always inspiring me to be true to myself. And I am eternally grateful to my loving partner, Anirudh, whom I met during my time at CMU. Thank you for being my rock and enabling me to chase my dreams, knowing that you will always be by my side.

# Contents


**1 Introduction** **1**
- 1.1 Motivation . . . 1
- 1.2 Thesis Goal and Contributions . . . 2
- 1.3 Thesis Organization . . . 4

**2 Related Work** **7**
- 2.1 Robot hands . . . 7
- 2.2 Benchmarks . . . 8
- 2.3 Design Iteration . . . 9
- 2.4 Real World Evaluation . . . 10

**3 Designing Delta Robot Manipulators** **11**
- 3.1 Introduction . . . 11
- 3.2 Related Work . . . 12
- 3.3 Approach . . . 13
  - 3.3.1 Characterization of Parallelogram . . . 13
  - 3.3.2 Evaluation of Delta Mechanism . . . 14
- 3.4 Experimental Results . . . 15
  - 3.4.1 Characterization experiments . . . 15
  - 3.4.2 Evaluation Experiments . . . 16
- 3.5 Conclusion and Future Work . . . 17

**4 Evaluating Delta Robot Manipulators** **19**
- 4.1 Introduction . . . 19
- 4.2 Related Work . . . 20
- 4.3 Delta Manipulator Design . . . 22
  - 4.3.1 Setup . . . 22
  - 4.3.2 Delta Actuators and Workspace . . . 22
  - 4.3.3 Fingertip design . . . 23
- 4.4 Delta Modeling . . . 24
  - 4.4.1 Offset between Revolute Joints . . . 24
  - 4.4.2 Learning Delta Robot Kinematics . . . 25
- 4.5 Force Profile . . . 26
- 4.6 Experiments . . . 28

4.7 Discussion and Conclusion . . . 29

**5 Designing and Evaluating Tendon-Driven Soft Hands 31**
5.1 Introduction . . . 31
5.2 Related Work . . . 33
5.3 Experiment Setup . . . 34
5.3.1 Robot Hand design . . . 34
5.3.2 Fabrication using 3D-printing . . . 34
5.3.3 Hand Evaluation using Teleoperation . . . 35
5.3.4 Manipulation Tasks . . . 37
5.4 DASH Iterative Design Studies . . . 38
5.4.1 Iteration V1 . . . 38
5.4.2 Iteration V2 . . . 39
5.4.3 Iteration V3 . . . 40
5.4.4 Iteration V4 . . . 41
5.4.5 Iteration V5 . . . 43
5.5 Baseline Study: Allegro Dexterous Hand . . . 44
5.6 Discussion . . . 44
5.7 Conclusion and Future Work . . . 45

**6 Evaluating Soft Hands using Learned Policies 47**
6.1 Introduction . . . 47
6.2 Related Work . . . 49
6.3 Fine-Tuning Affordance for Dexterity . . . 50
6.3.1 Learning grasping affordances . . . 50
6.3.2 Fine-tuning via Interaction . . . 52
6.4 Experiment Setup . . . 53
6.5 Results . . . 54
6.6 Discussion and Limitations . . . 56

**7 Automating Design of Soft Hands 59**
7.1 Introduction . . . 59
7.2 Related Work . . . 61
7.3 Dexterous Anthropomorphic Soft Hand . . . 61
7.4 Design and Policy Co-Optimization . . . 64
7.4.1 Problem Definition and Preliminaries . . . 64
7.4.2 Genetic Algorithm and Policy Transfer for Design Optimization . . . 64
7.4.3 Remarks . . . 65
7.5 Simulation Benchmark and Evaluation . . . 66
7.5.1 Benchmark Definition . . . 66
7.5.2 Evaluation Metrics . . . 67
7.5.3 Other Implementation Details . . . 67
7.6 Real-World Design Evaluation . . . 68
7.7 Discussion & Conclusion . . . 72

**8 Evaluating Automated Hand Design on DASH-30 Tasks** **73**
8.1 Automated hand design performance . . . 73
8.2 DASH-30 Tasks . . . 73
8.2.1 DASH v7 results on DASH-30 tasks . . . 73
8.2.2 DASH-v3 results repeated . . . 75
8.3 Discussion . . . 76

**9 Discussion** **79**
9.1 Research Questions . . . 79
9.1.1 Design Parameters . . . 79
9.1.2 Choosing Tasks . . . 82
9.1.3 Rapid Prototyping and Teleoperation Evaluation . . . 85
9.1.4 Simulated Hand Designs . . . 87

**10 Conclusion and Future Work** **91**
10.1 Conclusions . . . 91
10.2 Future Work . . . 91
10.2.1 Design Parameters . . . 91
10.2.2 Tasks . . . 92
10.2.3 Rapid Prototyping . . . 92
10.2.4 Teleoperation Evaluation . . . 92
10.2.5 Simulation . . . 93

**Bibliography** **95**

# List of Figures

1.1 Popular state-of-the-art (rigid) manipulators: (Left) two-finger gripper attached to a Franka robot arm [1] (Middle) Allegro dexterous hand [2] and (Right) ShadowHand [3]. . . . . 1

1.2 The two manipulators discussed in this thesis are (Left) Delta Robot Manipulators and (Right) Tendon-Driven Anthropomorphic Soft Hand. . . . . 3

1.3 Design iteration method versus evaluation method of compliant manipulators. Each of the completed works and proposed works are placed with respect to both of these axes. . . . . 4

2.1 Diagram of topics covered in thesis related work. The works presented in this thesis fall into the categories outlined in red. . . . . 7

3.1 (a) A delta robot with each arm consisting of a rigid portion and a flexible parallelogram with living hinges that transfers motion from motors at base to end-effector. (b) Ideal behavior of displaced tail of the parallelogram in characterization experiments testing different hinge and beam thicknesses and materials. . . . . 12

3.2 (a) Parallelogram characterization experiment setup: linear actuator mounted with compression load sensor to displace the tail of a PP parallelogram with beam thickness = $2.5$mm and hinge thickness = $0.375$mm. (b) Force-displacement plots for PP and (c) TPU for varying beam thicknesses (shown with shapes) and hinge thicknesses (shown with colors). Each marker represents the mean across five repetitions of the experiment. . . . . 14

3.3 Characterization experiment results for Z rotation angles (in degrees) of PP and TPU parallelograms at the largest displacement ($20$mm). The mean and standard deviation shown as error bars are taken over five repetitions of the experiment. The most compliant hinges ($0.375$mm) and least compliant hinges ($1.0$mm) are grouped together, and the colors denote different beam thicknesses of $2.5$, $3.5$, and $4.5$ mm. . . . . 16

3.4 Evaluation experiment parallelograms with beam thickness $2.5$mm and hinge thickness of (a) $1$mm and (b) $0.375$mm are shown before and after lowering one arm (2 arms up, 1 down) by $50$ degrees. . . . . 18

4.1 Novel robotic gripper grasping a coin, a card, and a dough roll with two delta robots, each with three degrees of freedom and made from 3D-printed soft material, polypropylene (PP). The linear actuators, compliant delta links, and fingertips are shown for a pair of delta robots. The delta robot's links move up and down with the linear actuators fixed at the joints. . . . . 20

4.2 This diagram illustrates the two orthogonal axes of rotation that approximate a universal joint and the measurements of $k, s_p$, and $L$. $k$ is shown as the distance between the two orthogonal axes of rotation, $L$ is shown as the distance between the axes of rotation at the top and bottom of a leg, and $s_p$ is shown as the distance between the the axes of rotation where the center of the parallelogram links attach to the end-effector. Each measurement represents a distance constraint between revolute joints in our simulation. . . . . 21

4.3 The prismatic delta workspace, in centimeters, has a dome-like shape, where the robot can reach any of the points in X, Y, and Z axes. The three vertical red lines represent the position of the linear actuators. Colors are mapped to the height for a better visualization. The actuator lengths are limited to $4$ cm so that no actuator can be above the delta's end-effector. Higher z values can be reached by adding a constant offset to every actuator. . . . . 23

4.4 Effects of varying the offset $k$ on the mean error $\mu$ from the path of a standard delta. The path of the delta with offsets is shown in red, and the path of a standard delta is shown in blue. For $k = .7$ cm, some actuator inputs become infeasible and the path is cut short. . . . . 24

4.5 (a) Marker-based stereo perception system to track the delta end-effector position using two orthogonal Logitech C920 cameras. The graphs compare measured and desired delta positions on the (b) XY and (c) XZ plane when following a test path (shown in blue) with the PP delta. The neural network trajectory prior to training is shown in red, and the trajectory after training is shown in green. . . . . 25

4.6 (a) Force profile experiment setup consisting of a GSO-500 Transducer Techniques Load Cell and TPU delta robot with planar fingertip. The delta robot pushes on the load cell with a displacement of $5$mm at various positions in the workspace, along X and Y axes. The mean force exerted by the delta at different values of x and y, and standard deviation are shown for the (b) PP deltas (c) TPU deltas. . . . . 26

4.7 Panels (a), (b), and (c) show the delta grippers with planar fingertips 3D-printed using PLA grasping a grape, aligning a pile of coins, and taking a coin from that pile and rotating it in hand, respectively. Panel (d) shows the deltas with spherical TPU fingertips picking a grape from its stem. . . . . 27

4.8 Timelapse of the compliant delta gripper sliding the top card and picking it up from the deck, and rolling a flat piece of dough into a spiral. . . . . 29

5.1 Manipulation task performance over five iterations of DASH designed through rapid prototyping and real-world evaluation on tasks alongside task performance of our baseline hand Allegro. . . . . 32

5.2 (a) Our soft robotic hand design process involving rapid prototyping and real-world evaluation (b) CAD models and differences across DASH iterations V1 through V5, as explained in Section 5.4. . . . . 33

5.3 (a) Assembly of DASH-V3 (top to bottom) including fingers, palm, top plate, motors, bottom plate, xArm6 mount. (b) Calibration procedure to map motor angles to finger joint positions, where tendon 0 actuates MCP side-to-side, tendon 1 actuates MCP forward folding motion, and tendon 2 curls the finger controlling both PIP and DIP joints. . . . . 35

5.4 Manus Meta Quantum Metagloves used for tracking the hand for teleoperating the robot arm and DASH. . . . . 36

5.5 Subset of tasks with different performance success across V1 to V5 on specific tasks used to inform design iteration. The top row of inset images shows representative tasks of successful tasks for each hand. . . . . 38

5.6 Task performance over 5 repetitions of each task across V1, V2, V3, V4, V5, and Allegro as baseline. The tasks are ordered difficult to easy from left to right, according to task performance of Allegro. . . . . 42

5.7 Task performance across V1, V2, V3, V4, V5, and Allegro as baseline on each category of tasks from Table 5.2. . . . . 44

6.1 **Left:** DEFT consists of two phases: an affordance model that predicts grasp parameters followed by online fine-tuning with CEM. **Right:** Our affordance prediction setup predicts grasp location and pose. . . . . 48

6.2 We produce three priors from human videos: the contact location (**top row**) and grasp pose (**middle row**) from the affordance prior; the post-grasp trajectory (**bottom row**) from a human demonstration of the task. . . . . 50

6.3 **Left**: Workspace Setup. We place an Intel RealSense camera above the robot to maintain an egocentric viewpoint, consistent with the affordance model's training data. **Right**: Thirteen objects used in our experiments. . . . . 52

6.4 Qualitative results showing the finetuning procedure for DEFT. The model learns to hold the spatula and flip the bagel after 30 CEM iterations. . . . . 53

6.5 Improvement results for 6 tasks: pick cup, pour, open drawer, pick spoon, scoop, and stir. We see a steady improvement in our method as more CEM episodes are collected. . 55

6.6 We evaluate DEFT on three difficult manipulation tasks. . . . . 55

7.1 Automated design iteration for dexterous soft hands: a) design optimization of robotic hand designs in simulation using genetic algorithms and policy transfer, b) fabrication of optimized hand that outperforms other hand designs in simulation, and c) design evaluation of optimized hand in the real-world using teleoperation on the same set of manipulation tasks. . . . . 60

7.2 a) CAD of top-performing optimized hand v7 shown with DIP, PIP, and MCP joints labelled, as well as hand design parameters shown with green arrows. Tendon placements along finger are shown in blue where each controls MCP adduction-abduction movement, MCP forward flexion, and curling the finger inwards through DIP and PIP flexion. The distal, middle, and proximal phalanges are also labelled in green to show some of the hand design parameters. b) Visualization of simulated top-performing hand design v7 . . . . . 62
7.3 Top three optimized hands. From left to right, the success rate AUC (%) of the hands are 53.32%, 53.07% and 52.47% respectively. . . . . 67
7.4 The 6 goal poses (shown for `pen` object) used for real world teleoperated manipulation tasks. . . . . 69

8.1 Tasks succeeded (out of 30 total) for v1, v2, v3, v4, v5, and v7, where the task succeeded for all five repetitions of the task. Each hand design iteration improved with the automated design iteration using simulation, v7 outperforming the previous designs and our baseline Allegro hand. Each hand did best on different tasks, as shown in the insets above the graph. . . . . 74
8.2 Task success on DASH-30 suite of tasks for 6 iterations of DASH including the baseline Allegro hand. Our automated hand design iteration, v7, improves on our manually designed iterations v1 to v5 and outperforms Allegro. . . . . 74
8.3 Task performance over 6 iterations of DASH on all DASH-30 tasks. The tasks are ordered difficult to easy from left to right, according to task performance of Allegro. . . . . 75
8.4 Task performance across DASH v1, v2, v3, v4, v5, v7, and Allegro as baseline on each category of tasks from Table 5.2. . . . . 75
8.5 Subset of tasks with different performance across DASH v1 to v7 on specific tasks. The top row of inset images shows representative tasks of successful tasks for each hand. . . . . 76

9.1 Top 25 optimized hands among 821 searched hands using design automation pipeline in Chapter 7. . . . . 82
9.2 Hand performance results on DASH-30 suite of tasks for Allegro baseline hand, v1, v2, v3, v4, v5, and v7 across manipulation categories hold, pick and place, lever, twist, open, and put in/on. Each task result is shown as the number of repetitions succeeded (legend in top right corner) out of five total repetitions of each task for each hand. . . . . 84

# List of Tables

1.1 Delta gripper and Dexterous Anthropomorphic Soft Hand (DASH) are the two manipulators discussed in this thesis. This table depicts the differences and specifications of each manipulator. Manipulator costs do not include the 3D printer's cost. . . . . 3

3.1 Characterization experiment results for X, Y, and Z rotation of parallelograms at 20mm displacement. The mean ($\mu$) and standard deviation ($\sigma$) is taken over five repetitions of the experiment. The last two columns consist of data from evaluation experiments integrating parallelograms of varying beam and hinge thicknesses and calculating the rotation of end effector (in degrees). The mean and standard deviation of three repetitions raising 2 arms and lowering one arm (2 up, 1 down) as well as lowering 2 arms and raising 1 arm (2 down, 1 up) by 50 degrees is listed in fourth and fifth columns respectively. . . . . 17

5.1 Calibration weights for all five iterations of DASH mapping from finger joint angles to motor angles. . . . . 36

5.2 DASH-30: task set of 30 manipulation experiments. Tasks with the asterisk (*) were hand-picked as tasks where compliance of the hand may be advantageous. . 37

5.3 Hand design parameters where finger length refers to the distances in millimeters from the top of the MCP joint to the fingertip and finger strength (N) is measured by pulling on a fully curled finger with a digital force gauge. . . . . 40

6.1 Parameters that are fine-tuned in the real world. The affordance model predicts a 45-dimensional hand joint pose for $P$, which is retargeted to a 16-dimensional soft hand pose. . . . . 51

6.2 We present the results of our method as well as compare them to other baselines: Real-world learning without internet priors used as guidance and the affordance model outputs without real-world learning.We evaluate the success of the methods on the tasks over 10 trials. . . . . 54

6.3 Ablations for (1) reward function type, (2) model architecture, and (3) parameter estimation. . . . . 56

7.1 Hand design parameter ranges tested in simulation to find the optimized robotic hand design where ff, mf, and rf refer to first (index) finger, middle finger, and ring finger, respectively. Unit is millimeter for lengths and degree for angles. Designs with * are optimized hand designs. . . . . 63
7.2 Teleoperation evaluation results for pick up and reorient tasks for each of the six objects on all goal poses using different hand designs. The above table shows both the average and max grasp quality score achieved. . . . . 70
7.3 Teleoperation evaluation results for pick up and reorient tasks for each of the goal poses on all six objects using different hand designs. The above table shows both the average and max grasp quality score achieved. . . . . 70
7.4 Simulation evaluation results for pick up and reorient tasks for each of the six objects at scale $1\times$ on randomized goal poses using different hand designs. V7-wedge is V7 with a wedge-like fingertip shape. The above table shows AUC of success rate as given by equation (7.2). . . . . 71

8.1 Results of experiments. Task success measures the cumulative number of tasks where the hand succeeded. For example, $4/5$ means at least four out of five repetitions of the task were successfully completed and $0/5$ means none of the repetitions were successful. . . . . 77

9.1 DASH hand design parameters compared to human hand proportions, where proximal, middle, and distal phalanx lengths were taken from middle finger measurements [4] and hand index is the palm width [5] divided by hand length as a percentage. All measurements are in millimeters. . . . . 82

# Chapter 1

# Introduction

## 1.1 Motivation

We envision the future of personal robotics to have robots in our homes helping us with everyday tasks. Ideally, these tasks would involve working with and around humans and interacting with delicate objects. Furthermore, to have these robots widely deployed we require these robots to be accessible at a low-cost to the consumer. However, modern off-the-shelf robot manipulators are expensive, difficult to manufacture and repair, and require extensive maintenance. As an example, consider the Shadowhand in Figure 1.1(right) which costs around $\$50000$ making it inaccessible to the average user. Furthermore, these manipulators have rigid components that limit their capability to work with delicate objects and around humans. For example, consider the household task of opening a plastic bottle cap. Robot hands with rigid parts can easily exert too much force, resulting in ejecting the bottle from the hand, and lack the dexterity required to complete this task.

Soft robot manipulators, on the other hand, exhibit compliance due to the use of soft materials which allows them to conform to objects and safely handle them. In addition, the high degrees of freedom also allow these manipulators to perform dexterous manipulations that is required in household tasks. Furthermore, rapid prototyping techniques like 3D-printing have made soft robots accessible and can be fabricated at a low cost unlike their rigid counterparts. However,

Figure 1.1: Popular state-of-the-art (rigid) manipulators: (Left) two-finger gripper attached to a Franka robot arm [1] (Middle) Allegro dexterous hand [2] and (Right) ShadowHand [3].

current deployment of soft robot hands is limited as their design process is extremely challenging and time-intensive. Modeling soft materials in simulation is computationally expensive as we need to model how these materials deform when subject to forces, making the use of simulation in design iteration difficult. Thus, it is challenging to iterate on the soft manipulator design prior to manufacturing. A natural question that arises from this is: *Can we improve the design iteration process for soft robot hands in order to quickly create a design, fabricate it, and test it in the real world to inform future iterations, all in a matter of days?*

This question is especially relevant at this stage of development of low cost robotic hands, because it allows the designer to quickly assess whether an iteration of the hand is going to be successful for the tasks we want it to do in the real world. By fabricating a prototype and testing it in the real world, the designer can use real-world measurable metrics to improve future soft hand designs. This results in a rapid turnaround in the design iteration process, while consistently improving the performance of the hand on real-world tasks.

This thesis takes an important step towards answering this question and explores various rapid design iteration approaches by designing and evaluating two different types of manipulators in the real world: a parallel delta manipulator and a tendon-driven anthropomorphic (human-like) hand.

## 1.2 Thesis Goal and Contributions

**Thesis statement:** *Achieving robot hand dexterity requires detailed attention to hand kinematics and compliance.*

This thesis explores techniques for iterative design including rapid prototyping, teleoperation, and simulation to enable designers to tune kinematics and compliance for dexterous tasks through real robot evaluation. We show that substantial improvements can be made between design iterations and over state of the art dexterous robotic hands on dexterity benchmarks. We employ 3D-printing techniques for rapid turnaround on fabricating soft hand designs, and the ability to quickly customize the design for future iterations. For evaluating our designs, we focus on real-world tasks to ensure that the final design parameters are optimized for task success, as opposed to task-agnostic metrics. By the end of the thesis, we describe the strengths and weaknesses of our approaches in terms of effectiveness, the time taken to design a prototype with a set of task goals in mind and to evaluate the prototype in the real world, ease of defining real-world measurable task metrics, and overall insights into choosing design parameters and evaluating tasks for hand design iteration.

In this thesis, we design and evaluate two manipulators, as shown in Figure 1.2: a 3D-printed compliant delta manipulator and a Dexterous Anthropomorphic Soft Hand (DASH), a tendon-driven hand. Delta manipulators are parallel manipulators that consist of links attached to the end-effector in parallel. Figure 1.2 (left) shows the compliant links that are structured as parallelograms that translate the end-effector platform in X, Y, and Z, giving each delta manipulator finger 3 degrees-of-freedom (DoF). Delta robots have been shown to have high precision and closed-form kinematic solutions, even at the millimeter-scale [6]. In addition to these advantages, we leverage compliance to do fine-grained precise manipulations with our delta manipulator. Figure 1.2 (right) shows our soft tendon-driven hand, DASH, which uses four tendons per

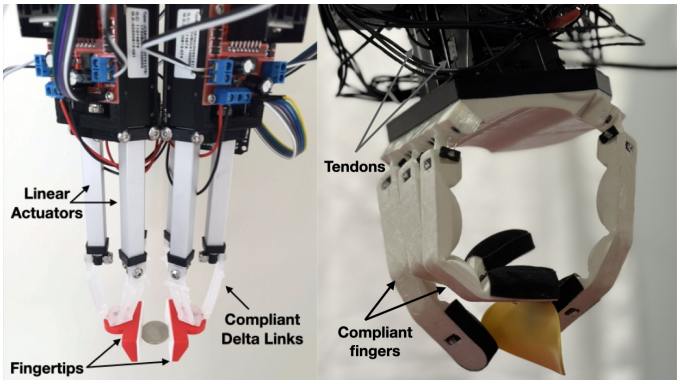



Figure 1.2: The two manipulators discussed in this thesis are (Left) Delta Robot Manipulators and (Right) Tendon-Driven Anthropomorphic Soft Hand.

| | Type of manipulator | Actuation | Fingers | DOF | Material | Cost |
|---|---|---|---|---|---|---|
| Delta | Parallel manipulator | Linear Actuators | 2 | 6 | TPU 95A | $300 |
| DASH | Anthropomorphic | Tendon-driven | 4 | 16 | TPE 83A | $1000 |

Table 1.1: Delta gripper and Dexterous Anthropomorphic Soft Hand (DASH) are the two manipulators discussed in this thesis. This table depicts the differences and specifications of each manipulator. Manipulator costs do not include the 3D printer's cost.

finger to contract the finger to close the hand, similar to a human hand. We use DASH to achieve more dexterous and contact-rich manipulations with all four fingers, and also to exert more force than was possible with the delta robot manipulator. Both manipulators are 3D-printed from soft materials and benefit from compliance to execute tasks safely with delicate objects. The specifications of each of the 3D-printed compliant manipulators are detailed in Table 1.1. We explore different design iteration and evaluation methods for these manipulators as described in our contributions below.

We summarize our thesis research in Figure 1.3 and list them below:

- Designing Delta: Designed compliant 3D-printed parallelogram links for delta robot manipulators, leading to the development of the first 3D-printable compliant delta robot using soft materials. *24 designs evaluated in real world*
- Evaluating Delta: Evaluated a novel gripper composed of two 3-DOF compliant delta robots, which utilized the designed parallelogram links, using teleoperation in real-world tasks. *1 design evaluated in real world*
- Designing and Evaluating DASH: Developed a framework for designing anthropomorphic soft hands using 3D printing-aided fast prototyping and teleoperation, which resulted in the creation of the Dexterous Anthropomorphic Soft Hand (DASH). *5 designs evaluated in real world*
- Evaluating DASH: Learned policies from internet videos and online fine-tuning on dex-

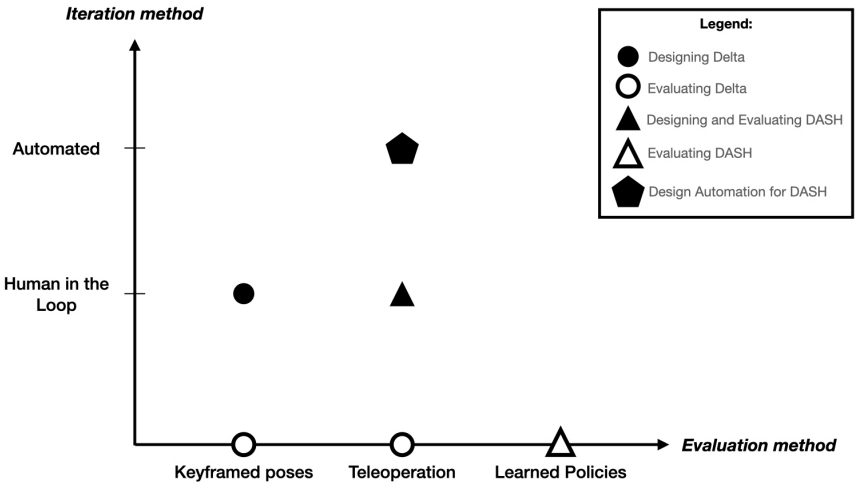



Figure 1.3: Design iteration method versus evaluation method of compliant manipulators. Each of the completed works and proposed works are placed with respect to both of these axes.

terous tasks, instead of teleoperation. *1 design evaluated in real world*

- Design Automation for DASH: Co-optimizing design and control of dexterous hands for automated design iteration of soft hands in simulation and using real-world teleoperation evaluation, instead of manual design iteration. *2 designs evaluated in real world*

## 1.3 Thesis Organization

In Chapters 3 and 4, we design and evaluate a compliant 3D-printed delta robot. First, we design the delta robot using keyframed poses in characterization experiments and subsequently, evaluate the delta robots using a combination of keyframed poses and teleoperation, as shown in Figure 1.3. The design process here is considered human-in-the-loop since it relies on adjusting the design parameters using the expertise of the designer based on the performance during evaluation. Chapter 3 focuses on the characterization of the compliant parallelogram links used to create the delta robots. Subsequently, Chapter 4 evaluates the design of these 3D-printed compliant delta robots as fingers in a 2-finger robotic gripper to do low-inertia manipulations. While our delta robot designs were able to achieve difficult tasks like rolling dough and picking a grape off of a stem, we look to designing tendon-driven soft hands for compliance and dexterity, as well as increased strength.

In Chapter 5, we describe a framework for rapid design iteration to build and iteratively test soft robot hand designs on real-world manipulation experiments. Typically the process of building, testing, and re-designing robotic hands is costly and time-consuming [7]. Rapid prototyping techniques such as 3D-printing and teleoperation are sufficiently advanced to support a process of rapid design iteration. Our framework leverages these techniques to rapidly evaluate soft hands on real-world manipulation tasks in order to inform subsequent hand designs. As shown in Figure 1.3, we use human-in-the-loop design iteration and use teleoperation to evaluate the hand's performance on a real-world manipulation task suite named DASH-30. We create five

iterations of dexterous anthropomorphic soft hand (DASH), improving task performance incrementally over each iteration. Two of our design iterations required only 3 and 5 days to design and test, thereby showing that we can build, test, and improve on soft hand designs quickly.

In contrast to our keyframed and teleoperation approaches, in Chapter 6, we leverage both human demonstration videos and online fine-tuning to learn policies that perform complex tasks with DASH, such as flipping a bagel. The goal of this work is to demonstrate the capability of DASH on complex dexterous tasks autonomously. Additionally, this opens a new modality of evaluation in our framework, besides teleoperation. Our final DASH iteration in Chapter 7 focuses on automating the design iteration process, thereby taking the human out of the loop, in order to explore a larger design space. To achieve this, we simulate the soft hand as a rigid body and use hand evolution methods to transfer control policies across a large set of generated hand designs to fabricate and determine the best design based on the success rate on simulated dexterous tasks. Subsequently, we fabricate the design and teleoperate it on real-world tasks to both evaluate its performance. In Chapter 8, we test the optimized hand designs on our suite of tasks DASH-30 to evaluate the generalization of their capabilities. Finally, Chapter 9 discusses the findings from our design iteration approaches, and Chapter 10 concludes our work and considers future extensions of our work.

# Chapter 2

# Related Work

In existing work, dexterous hands generally look like multi-fingered hands or anthropomorphic morphologies. This alludes to human-like dexterity through biomimetic approaches or heuristic approaches to achieve dexterous capabilities. Many of these hands are tested on a limited set of tasks or experiments to demonstrate these capabilities. These benchmarks can look like suites of tasks or taxonomies. During evaluation on these tasks, we can think about how to increase dexterity. We can learn strategies to increase the capabilities of existing hands using concepts such as extrinsic dexterity. Or we can improve intrinsic dexterity through improved hardware. We focus on the latter in order to build a good foundation to build on using software. In this thesis, we create hands efficiently with newer rapid prototyping techniques and evaluate them faster than previously with faster simulations and real-world evaluation methods. Simulating hand designs allows us to test designs before making them but advances in fabrication techniques allow us to use both for a closed-loop design iteration method.

## 2.1 Robot hands

Robotic hands can be categorized as non-dexterous hands and dexterous hands, where dexterous hands are mostly characterized by multi-fingered hands. Multiple fingers are required to have the dexterity to do tasks such an rotating dice in-hand. However, related work shows that tasks

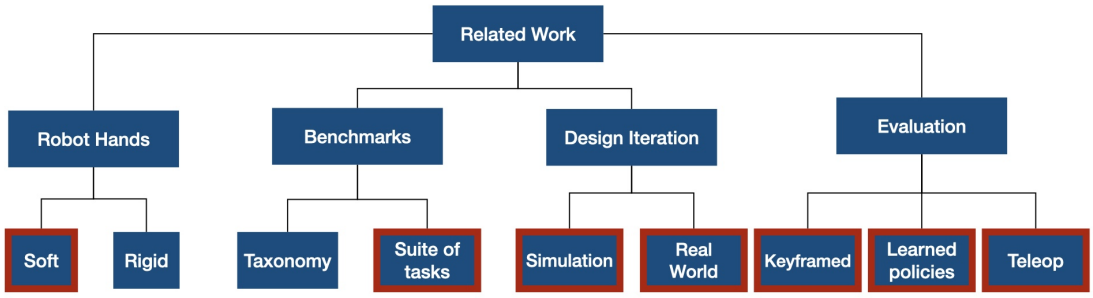



Figure 2.1: Diagram of topics covered in thesis related work. The works presented in this thesis fall into the categories outlined in red.

like bin picking (picking up a single object from a bin of many objects) can be done with suction grippers and parallel jaw grippers [8]. Non-dexterous hands are also capable of doing dexterous or complex manipulation tasks by taking advantage of extrinsic dexterity [9]. This concept requires using external contacts such as grasping an object into a table or using the table to flip an object, where the table acts as another contact instead of a finger. Extrinsic dexterity allows simple hands and grippers to achieve more complex tasks but they cannot efficiently solve tasks, e.g., rotating dice in-hand. Without extrinsic dexterity, the gripper would place the dice on a table and pick up the dice in a new position (if that grasp is feasible). With extrinsic dexterity, this could involve letting the dice make contact with a surface and allowing the dice to rotate while being grasped. For a dexterous hand, this could be achieved by twisting the dice with the palm facing up using the thumb and remaining fingers. Simple grippers can use reinforcement learning and extrinsic dexterity to solve complex manipulation tasks including "occluded grasps" where the desired grasp is not feasible [10]. However, dexterous hands can efficiently solve complex tasks and human hands are an available heuristic for modeling dexterous hands.

In the last century, multi-fingered dexterous hands have seen a trend of moving from rigid robot hands to using soft joints and materials [11]. Popular rigid dexterous hands include ShadowHand [3] and Allegro [2], which are commercial robot hands. Rigid dexterous hands from research labs include D'Manus [12], LEAP [13], D'Claw [14], and many more. Similarly, existing soft hands range from tendon-driven foam hands [15] to pneumatic soft hands [16] to prosthetic hands [17]. In this thesis, we create compliant dexterous manipulators using soft materials and soft joints due to their safety in interactions with objects and environments [18]. Compliance allows for a different paradigm compared to rigid manipulation where we can opt for contact-rich manipulations rather than avoiding contact [11]. For example, when picking up a cloth from the table, a rigid manipulator would have to be careful not to jam into the table while grasping whereas the soft manipulator can slide along the table without damaging the hand, the table, or the object. We see that many of the existing dexterous hands are anthropomorphic (human-like) which drives us to design hands closer to the human hand in hopes of reaching human-like dexterity and efficiency in solving manipulation tasks. Similarly, we look at a two-finger gripper and then an anthropomorphic four-fingered robotic hand, both made from soft materials and joints.

## 2.2 Benchmarks

In addition to designing robot hands, designing benchmarks to compare and evaluate designs is important to understand which tasks certain designs are better suited for. When looking at dexterous manipulations, existing taxonomies try to categorize everyday grasps or correlations to human hand poses. These taxonomies are not about performing specific manipulations on different objects, but rather about capturing the limitations and capabilities of the hand itself by observing static hand poses. The GRASP taxonomy [19] incorporates grasp classifications of various objects sizes and shapes including power, intermediate, and precision grasps along with even more granular subcategories. Another popular taxonomy is the Kapandji score [20] which does not use any objects but focuses on thumb opposition and reachability to 10 locations on remaining fingers and the palm. Grasp taxonomies can also be presented as task lists based on everyday grasps studied from human participants [21], which can inform us on the variety of

tasks we would like robot hands to be able to perform.

While taxonomies can evaluate whether static hand poses are feasible, benchmarks such as the Elliott and Connolly Benchmark [22] present a suite of dynamic manipulation patterns to evaluate in-hand dexterity and ability to move between grasp poses of specific objects. Benchmarks containing suites of tasks that involve various objects such as Purdue Pegboard Test [23], Box and Blocks Test [24], Jebsen Taylor Hand Function Test [25], and SHAP Test [26] evaluate the ability to complete daily activities or assess rehabilitation of manual dexterity. In additions to suites of tasks, there are benchmarks of objects to test how a robot hand would grasp a variety of shapes and sizes of objects (same grasping task but different objects). Datasets such as YCB [27] allow many robot hand designs to attempt grasping the same object dataset to compare hand design performances with existing hand designs. Similarly, suites of tasks and objects in simulation, such as DAPG [28], provide benchmark environments and manipulation tasks such as door opening, hammering a nail, pick and place, and in-hand pen reorientation to evaluate manipulation learning methods. In this thesis, we take inspiration from everyday grasps [21] and in-hand dexterity benchmarks [29, 30, 31] to curate our own suite of tasks with an intention to evaluate beyond grasping or pick-and-place tasks.

## 2.3 Design Iteration

Design optimization and iteration involves testing different designs in either simulation or the real world in order to find designs that can perform well in the real world. While simulating robot designs can provide insights into robot performance without the time and effort of manufacturing and testing each design in the real world, the final designs are eventually tested in the real world prior to deployment. Thus, it is imperative that design iteration be grounded in real world performance. In locomotive robots, we see many designs tested in simulation and then testing one or more designs in the real world for both rigid robots [32] and soft robots [33]. Evaluating top performing designs in simulation in the real world verifies the design performance in simulation and reveals the sim to real gap. Domain randomization such as varying terrain for locomotive robots can lessen the sim to real gap [34].

While manipulation does not have an equivalent of tough terrain to test many designs on, existing work has explored computationally and heuristically. Heuristic approaches generally involve kinematic testing prior to fabricating a single design in the real world [7, 35, 36, 37, 38]. On the other hand, computationally tested hands require optimization of design or co-optimizing design and control for picking one design over many. Such approaches evaluate many designs in simulation in order to search a larger design space than iterating in the real world, where either one of the designs is replicated in the real world [15, 39, 40, 41] or results are purely in simulation [42, 43, 44]. However, since making many designs in the real world to evaluate is tedious and difficult, our work aims to fill this gap by rapidly iterate on designs in the real world with rapid prototyping and teleoperation. Additionally, we use co-optimization of hand design and control in simulation to evaluate designs, and then we fabricate multiple designs in the real world to verify them.

## 2.4 Real World Evaluation

Hand designs must be fabricated and evaluated in the real world for downstream manipulation task performance. In order to thoroughly evaluate a robot hand's capabilities, evaluation methods require both fabrication and control methods. Manufacturing methods for robot hands include any combination of machining [2], casting [45], folding [6], 3D-printing [46], etc. Soft pneumatic hands like RBO Hand 3 [16] require casting and curing and can take five days to completely build. However, recent advancements in 3D-printing technologies have made it an efficient rapid prototyping technique even for soft robots [47]. Thus, we leverage 3D-printing in our works to rapidly prototyping hand design iterations in the real world, sometimes designing, fabricating, and evaluating hands in under five days.

Control methods can vary from curated hand poses or grasps to autonomous trajectories or manipulation sequences. Keyframed poses can verify kinematic capabilities of the hand whether that informs thumb placements for grasping [7] or showcases the soft dexterous hand design's capabilities through Kapandji and GRASP taxonomy scores [48]. RBO Hand 3 (2022) [16] improves on both Kapandji and GRASP taxonomy scores compared to RBO Hand 2 (2016) [48]. Teleoperation can be used to learn mappings between human hand poses and robot hand poses [15] or evaluate manipulation tasks a hand can do [13]. Reinforcement learning methods can help create learned policies for manipulation tasks such as rotating a valve [49] or rotating objects in hand [50]. All of these are valid methods to test real world hand design performance, however autonomous policies are the end goal for robots to be deployed into shared environments with people. We use combinations of keyframed poses, teleoperation, and learned policies to iterate on the design of soft dexterous robot hands.

# Chapter 3

# Designing Delta Robot Manipulators

**Paper:** *Mannam, P., Kroemer, O., & Temel, F. Z. (2021). Characterization of compliant parallelogram links for 3D-printed delta manipulators. In Experimental Robotics (ISER): The 17th International Symposium (pp. 75-84). Springer International Publishing.*

## 3.1 Introduction

Personal care robots operating in homes require dexterous and compliant manipulators to ensure that interactions with human and objects are performed reliably and safely. Delta robots, Figure 3.1(a), are ideal for these applications due to their low-inertia and high-precision [51]. The dexterity and precision of these robots enable them to manipulate objects in unstructured home environments. However, most delta robots are made from rigid materials [52] which precludes them from interacting safely with humans. The compliance of soft materials, on the other hand, can allow them to conform to objects, exhibit physical robustness, and execute safe human interactions [18]. Furthermore, 3D-printing delta robots using soft materials reduces fabrication costs and makes them accessible to a wider community. This paper presents the process of designing the first 3D-printable compliant delta robot using soft materials. We characterize parallelograms, an integral component of the delta robot, with varying dimensional parameters and materials to study their behavior as part of a delta mechanism. More specifically, we focus on parallelogram hinge and beam thicknesses, and material stiffness. These design choices can then be leveraged to achieve desired end-effector behaviors of 3D-printed delta robots composed of compliant parallelogram links.

Printing compliant delta robots requires a departure from the classical delta design due to the use of soft materials, especially with respect to the joints. Rigid universal joints are replaced by two types of living hinges[1], one in the compliant parallelogram links of the delta robot and the other between the end-effector and the parallelogram links, as seen in Figure 3.1(a). Properties inherent to soft materials like bending and hysteresis can alter the delta mechanism behavior. As shown in Figure 3.1(a), we expect the end-effector (or platform) of the delta robot to always stay parallel to its base as seen in the upright position; however, this orientation is not guaranteed

[1] A **living hinge** is an articulated joint created by locally reducing the width of the material to make it more compliant in that region.

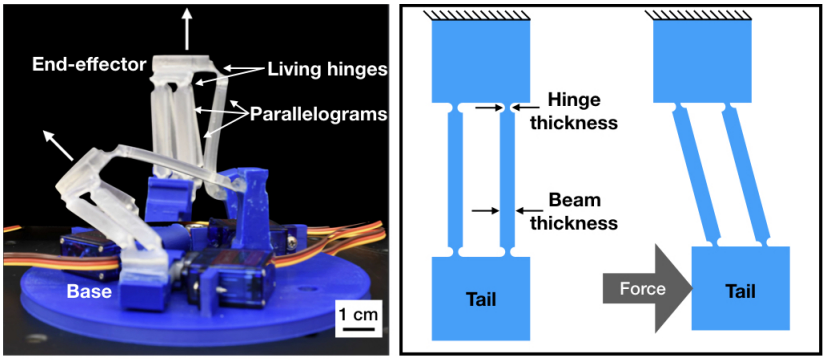



Figure 3.1: (a) A delta robot with each arm consisting of a rigid portion and a flexible parallelogram with living hinges that transfers motion from motors at base to end-effector. (b) Ideal behavior of displaced tail of the parallelogram in characterization experiments testing different hinge and beam thicknesses and materials.

in all configurations when using compliant materials. This deviation from the ideal kinematic behavior can be measured by the end-effector rotation with respect to the base across the robot configuration space. In this work, we study the characteristics of centimeter-scaled compliant parallelograms with 12 varied combinations of hinge and beam thicknesses, and 2 materials by displacing them incrementally and observing the force required and resulting rotation of the parallelogram tail, as shown in Figure 3.1(b). Subsequently, we evaluate these parallelograms by integrating them into the delta robot structure and measuring the end-effector rotation with respect to the base in specific configurations of the robot. These experiments allow us to infer the parameters of the compliant parallelogram link that achieve the ideal kinematic behavior, bringing us one step closer to the goal of creating a 3D-printed compliant delta manipulator.

## 3.2 Related Work

Compliant robots have been used in biomedical applications and are built on micro or milli scale using flexure-based mechanisms. Manufacturing compliant robots on such a small scale is difficult using traditional techniques and are typically built using either Micro Electro Mechanical Systems (MEMS) [6, 53, 54], or Electro-Discharge Machining (EDM) [55, 56, 57, 58]. These manufacturing techniques require sophisticated tools that are not accessible to the wider community. An alternative approach is to utilize origami-inspired self-folding robots [59, 60, 61] or laminate robot mechanisms [6, 62, 63]. Our work is closest to the millimeter-scaled Delta robot using laminate structures [6]. Unlike the manual assembly required for laminate robot mechanisms, we look to successful prototyping of compliant mechanisms using 3D-printing [64]. Our goal is to create a low-cost compliant delta manipulator without using complex manufacturing techniques by relying on 3D-printing.

A popular alternative parallel manipulator is the 6-DOF Stewart platform [65] that can translate and tilt the end-effector unlike the delta robot which only allows the end-effector to translate parallel to the base. Both of these manipulators can move at high speeds and accelerations with

low-inertia because their relatively heavy motors are stationary and at the base. Delta robots, unlike Stewart platforms, have simple closed-form kinematic solutions [52]. The characteristics improve as the mechanisms are scaled down, which is why delta robots like the mm-scaled delta (milliDelta) can move with precision down to $5$ micrometers in a $7\text{mm}^3$ workspace and lend themselves to micromanipulation [6]. Previous mm-scaled delta robots required multi-step processes or manual assembly which introduced irregularities in the final platforms [6][63]. In contrast, we use 3D-printing to make repeatable and accessible delta robots.

## 3.3 Approach

The parallelogram structure plays an integral role in the delta robot's functionality [51]. The key components of the parallelogram are the beam and hinge thicknesses, as labeled in Figure 3.1(b). Hence, we start with characterizing the parallelograms with varying beam and hinge thicknesses, as well as two different materials, by recording the effect on force and position when displacing the tail of the parallelogram as shown in Figure 3.1(b). Subsequently, we study the compounding effects of three parallelogram links in a delta robot structure on the resulting end-effector orientation. This enables us to model the mechanical properties of different materials and dimensional parameters of the compliant parallelogram that result in motions close to the idealized delta mechanism.

We characterized and evaluated the parallelogram links over twelve combinations of dimensional parameters: 3 beam thickness values of $2.5$mm, $3.5$mm and $4.5$mm, and hinge thickness values of $0.375$mm, $0.5$mm, $0.75$mm, and $1$mm to study deviations from the ideal delta kinematic model caused by different hinge and beam thicknesses. For our experiments, we used Ultimaker S5 3D-printer and filaments. We chose the two most compliant Ultimaker filament materials which are best suited to living hinge mechanisms, polypropylene (PP) and thermoplastic polyurethane (TPU) 95A. For comparison, the tensile modulus is 220 MPa (using ISO 527) and 26 MPa (using ASTM D638) for PP and TPU respectively. Additionally, the flexural strength is 13 MPa and 4.3 MPa (both measured using ISO 178) for PP and TPU respectively. These values indicate that TPU is more compliant than PP, which is beneficial for the living hinges while the stiffness of PP is useful to maintain rigidity of the parallelogram beams.

### 3.3.1 Characterization of Parallelogram

The two parameters for characterization experiments were the parallelogram beam and hinge thicknesses (Figure 3.1(b)). The remaining dimensions were fixed, such as the length and depth of the beams which were $27$mm and $4$mm respectively, and the length of the hinge was always $2$mm. Compliant parallelograms were made from PP and TPU while supporting and mounting materials were 3D-printed using rigid polylactic acid (PLA) material. Characterization experiments involved mounting these parallelograms vertically and using a linear actuator to push the parallelogram tails in $5$mm increments to measure the exerted force, and resulting displacement and orientation change of the tails. We vertically mounted parallelograms to reduce the effects of gravity in pulling the parallelogram out of plane. As shown in Figure 3.2(a), an Actuonix L12-R Micro Linear Servo is mounted with a Transducer Techniques GS0-150 load cell to measure the

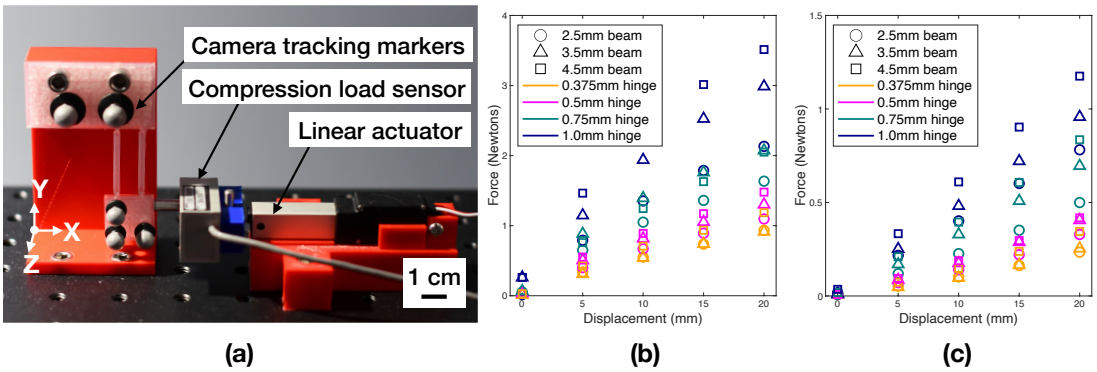



Figure 3.2: (a) Parallelogram characterization experiment setup: linear actuator mounted with compression load sensor to displace the tail of a PP parallelogram with beam thickness = 2.5mm and hinge thickness = 0.375mm. (b) Force-displacement plots for PP and (c) TPU for varying beam thicknesses (shown with shapes) and hinge thicknesses (shown with colors). Each marker represents the mean across five repetitions of the experiment.

force required to displace the parallelogram tail. For PP parallelograms with beam thickness of 4.5mm, we used the GS0-500 load cell instead to measure the higher force values that were required to displace these parallelograms. Three grey markers on the parallelogram are tracked as a rigid body by an OptiTrack V120:Trio camera with submillimeter accuracy to record the position and orientation of the parallelogram tail. The load stem was lubricated to overcome frictional effects when pushing the parallelogram. Force measurements were recorded after a six minute delay to allow for stabilization of any creep (deformation) effect. Additionally, rotation data from the markers on the parallelogram indicate if the tail of the parallelogram rotated out-of-plane or twisted in plane.

Using the same reference axes as shown in Figure 3.2(a), we observe in-plane and out-of-plane rotations of the parallelogram tail. The force applied by the actuator causes the tail of the parallelogram to move in an arc in the plane of the parallelogram. Under ideal conditions, the tail would remain parallel to the base. However, due to non-ideal hinge and beam behaviors, the tail exhibits additional rotations within the plane of movement, i.e., about the Z axis. Additionally, the restorative force of the parallelogram (caused by gravity and joint stiffness) will push back against the actuator. Misalignment between this restorative force and the actuator's pushing force can result in the parallelogram twisting (Y axis rotation) or bending out of plane (X axis rotation). Thus, we only focus on the in-plane rotations to test whether the parallelogram tail stays parallel to the mounted end of the parallelogram.

### 3.3.2 Evaluation of Delta Mechanism

Rotations of the compliant parallelogram links directly relate to the rotations of the delta robot end-effector. Hence, using the same parallelograms tested in the characterization experiments, we validated the results by integrating the parallelograms as links in an actuated delta robot as shown in Figure 3.4. The parallelogram links have an added living hinge on each side, oriented 90 degrees apart from the compliant links' revolute joint, to mimic one universal joint at each end

of the beams as shown in Figure 3.1(a). Three of these parallelograms are attached to a triangular end-effector platform, all created with either PP or TPU throughout. Then, the compliant delta tops are secured to 3D-printed rigid links made with Ultimaker Tough PLA (shown in red in Figure 3.4) which are attached to servo motors.

To measure the end-effector rotation with respect to the base at different configurations of the robot, we define two configurations for each of the arms, namely up and down, that are separated by a $50$ degrees offset. We consider two configurations of the robot, one in which two arms are down and the other arm is up, and the other in which two arms are up and the other arm is down. The three marker points on the end-effector are used to calculate the end-effector plane and normal axis. The normal axis is computed at each position of the delta robot, such that the angle between the normal vector and a reference base normal is used to compute the rotation of the end-effector with respect to the base.

## 3.4 Experimental Results

### 3.4.1 Characterization experiments

We tested parallelograms of varying hinge thickness ($0.375$, $0.5$, $0.75$, and $1$mm) with beam thickness of $2.5$mm, $3.5$mm, and $4.5$mm. The force values at the end of each $5$mm increment push until $20$mm for both PP and TPU, averaged over five repetitions, are shown in Figure 3.2(b) and (c) respectively. The circle, triangle, and square markers denote $2.5$mm, $3.5$mm, and $4.5$mm respectively, and each color denotes a difference hinge thickness between $0.375$ and $1$ mm. Five repetitions were executed for a given set of dimensional parameters and material on a newly printed parallelogram, resulting in a total of $120$ characterization experiments. There is a clear trend for both materials where displacing the parallelogram by a larger distance takes more force. Overall, PP parallelograms require more force to displace than ones made from TPU, needing almost $4$N versus close to $1.5$N for the stiffest (thickest) hinges at $20$mm displacement.

Using OptiTrack markers, Z rotation angles of the parallelogram tail were recorded after displacements of $5$mm increments up to $20$mm. In Figure 3.3, the mean and standard deviation for Z rotation angles are shown for both PP and TPU at the largest ($20$mm) displacement. For the same beam thickness, increasing the hinge thickness tends to result in larger Z rotations across both materials implying that in-plane rotations are smaller for thinner hinges. We can also observe that for a given beam and hinge thickness, TPU results in larger in-plane rotations when compared to PP. This can be attributed to higher compliance of TPU when compared to PP. Interestingly, for a fixed hinge thickness varying the beam thickness does not show consistent behavior across materials and hinge thicknesses. We leave exploring the reasons to future work.

The X, Y, and Z rotation angles with mean and standard deviation at $20$mm displacement of the parallelogram can be found in the first three columns of Table 3.1. As expected from our discussion in Section 3.3.1, X and Y rotations, corresponding to out-of-plane rotations, have high standard deviations due to misaligned restorative and actuator pushing forces. We have included these rotation values for completeness in Table 3.1. We can conclude that for both PP and TPU materials, a hinge thickness of $0.375$mm results in the least in-plane rotation. For beam thickness, the table indicates that both $2.5$mm and $4.5$mm for PP material have small in-plane

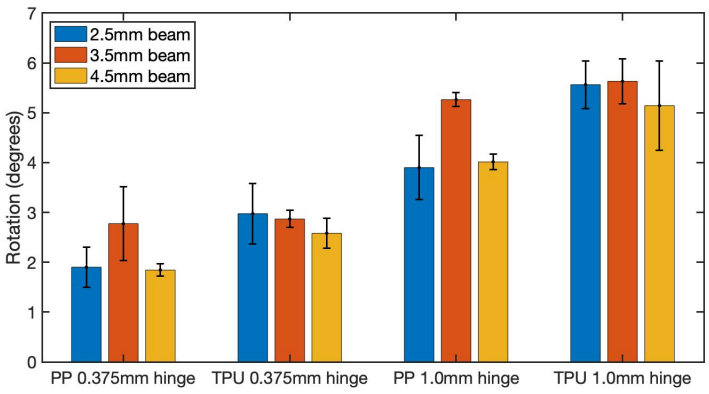



Figure 3.3: Characterization experiment results for Z rotation angles (in degrees) of PP and TPU parallelograms at the largest displacement (20mm). The mean and standard deviation shown as error bars are taken over five repetitions of the experiment. The most compliant hinges (0.375mm) and least compliant hinges (1.0mm) are grouped together, and the colors denote different beam thicknesses of 2.5, 3.5, and 4.5 mm.

rotations. Similar to Figure 3.3, we can infer that PP material achieves smaller mean in-plane rotation when compared to TPU material across all settings which can be again attributed to the difference in stiffness between the materials.

### 3.4.2 Evaluation Experiments

We evaluated the effects of integrating parallelograms from the characterization experiments, TPU and PP parallelograms with the most and least compliant hinges (0.375mm and 1.0mm), into full delta mechanisms. Specifically, we studied the effect on the end-effector orientation, relative to the base, when it is moved towards the edge of the configuration space, by lowering or raising each arm by 50 degrees, as shown in Figure 3.4. Both configurations were repeated three times with each of the three arms. We present the mean and standard deviation of the end-effector rotation in Table 3.1 (last two columns.) For both PP and TPU materials, hinge thickness of 0.375mm results in a smaller end-effector rotation when compared to a hinge thickness of 1.0mm across all settings. This highlights the importance of compliance in living hinges to achieve delta behavior that is closer to ideal. With a thick hinge, parallelogram links are stiff and unable to maintain the end-effector orientation to be parallel to the base. This is similar to the conclusion drawn from our characterization experiments where we observed that smaller hinge thicknesses are better for minimal in-plane rotations.

Among parallelograms with hinge thickness of 0.375mm, a beam thickness of 2.5mm works best for PP while a beam thickness of 4.5mm works best for TPU. This can be attributed to the stiffness of PP which requires thinner beam to be flexible enough to maintain the end-effector orientation across different configurations. On the other hand, TPU is more compliant and requires thicker beams to ensure that the beams stay parallel to each other which is important to achieve an end-effector orientation that is parallel to the base.

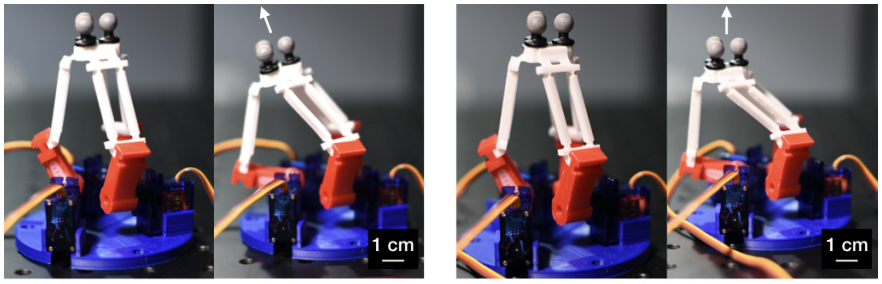
| (Beam th., Hinge th.) | X-rotation | | Y-rotation | | Z-rotation | | 2 up, 1 down | | 2 down, 1 up | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| PP (2.5, 0.375) | 2.31 | 2.11 | 2.28 | 0.85 | **1.90** | 0.41 | **6.27** | 1.68 | **5.65** | 1.66 |
| PP (3.5, 0.375) | 1.03 | 0.38 | 2.35 | 1.85 | 2.78 | 0.74 | 8.12 | 3.80 | 5.87 | 1.49 |
| PP (4.5, 0.375) | 2.08 | 0.84 | 4.68 | 1.56 | **1.84** | 0.12 | 8.75 | 2.61 | 6.15 | 3.83 |
| PP (2.5, 1.0) | 1.20 | 1.11 | 3.10 | 2.65 | 3.90 | 0.64 | 20.08 | 5.91 | 17.01 | 3.31 |
| PP (3.5, 1.0) | 1.04 | 0.38 | 1.20 | 0.57 | 5.26 | 0.14 | 15.23 | 0.86 | 13.22 | 1.69 |
| PP (4.5, 1.0) | 3.87 | 1.62 | 9.32 | 3.37 | 4.01 | 0.16 | 15.01 | 0.72 | 11.57 | 4.04 |
| TPU (2.5, 0.375) | 2.26 | 1.39 | 2.06 | 1.70 | **2.97** | 0.61 | 7.84 | 3.51 | 7.69 | 2.23 |
| TPU (3.5, 0.375) | 1.21 | 1.03 | 1.41 | 0.72 | **2.87** | 0.17 | 5.5 | 0.24 | 5.44 | 1.48 |
| TPU (4.5, 0.375) | 1.06 | 1.42 | 1.60 | 0.95 | **2.58** | 0.30 | **4.72** | 1.42 | **5.00** | 2.09 |
| TPU (2.5, 1.0) | 0.62 | 0.24 | 5.86 | 3.82 | 5.56 | 0.48 | 17.18 | 4.83 | 13.55 | 2.00 |
| TPU (3.5, 1.0) | 0.73 | 0.59 | 2.03 | 1.83 | 5.63 | 0.45 | 18.54 | 5.95 | 15.35 | 4.36 |
| TPU (4.5, 1.0) | 0.93 | 0.26 | 2.50 | 1.96 | 5.14 | 0.90 | 20.49 | 9.99 | 14.67 | 2.08 |

Table 3.1: Characterization experiment results for X, Y, and Z rotation of parallelograms at 20mm displacement. The mean ($\mu$) and standard deviation ($\sigma$) is taken over five repetitions of the experiment. The last two columns consist of data from evaluation experiments integrating parallelograms of varying beam and hinge thicknesses and calculating the rotation of end effector (in degrees). The mean and standard deviation of three repetitions raising 2 arms and lowering one arm (2 up, 1 down) as well as lowering 2 arms and raising 1 arm (2 down, 1 up) by 50 degrees is listed in fourth and fifth columns respectively.

The amount of permissible rotation is dependent on the application, but for manipulating large objects (centimeter scale or larger) 7 degrees of rotation or less is manageable. From a visual perspective as well, about 7 degrees of rotation in Figure 3.4(b) still demonstrates near-delta robot behavior.

## 3.5 Conclusion and Future Work

We tested 3D-printed parallelograms made with two materials of significantly varying stiffness, PP and TPU. The characterization experiments presented in Section 3.4.1 studied the effects of displacing the parallelogram tail. Due to the large variation in amount of force required for each of the parallelogram dimensions and materials, the force profile can be chosen for a specific application. For example, a delta robot that is interacting with humans for applications like assisted feeding is expected to be more compliant and therefore require less force to be displaced. Thus, we can tailor the parameters and material of the parallelogram links to the desired application




(a) TPU hinge thickness of 1 mm (b) TPU hinge thickness of 0.375 mm

Figure 3.4: Evaluation experiment parallelograms with beam thickness $2.5$mm and hinge thickness of (a) $1$mm and (b) $0.375$mm are shown before and after lowering one arm ($2$ arms up, $1$ down) by $50$ degrees.

using the force profile. Our characterization experiments have also indicated that more compliant hinges and PP material are better for achieving minimal in-plane rotations. Our evaluation experiments in Section 3.4.2 also show that more compliant hinges result in end-effector orientations that are closer to ideal delta behavior. In addition, our evaluation experiments have also provided insight into the ideal beam thickness for PP and TPU materials that result in minimal end-effector rotation.

Working with soft materials is challenging as their properties change over time and across repetitions. In order to control the 3D-printed delta for manipulation, we will have to learn how to accommodate for the material properties and adapt to changes. Our work is an initial step in this direction and the experiments show the potential of a 3D-printed delta using soft materials. Future extensions of this work involve manipulation tasks using opposing delta manipulators, arrays of delta robots for distributed actuation platforms, and modeling compliant delta robots. Modeling the delta robots also involves studying the error from rigid delta kinematic models and how to compensate for the error using robust control and state estimation techniques.

Next, we use this characterization of parallelograms for delta robots to design two delta manipulators opposing each other in a two-finger gripper configuration.

# Chapter 4

# Evaluating Delta Robot Manipulators

**Paper:** *Mannam, P., Rudich, A., Zhang, K. L., Veloso, M., Kroemer, O., & Temel, Z. (2021, July). A low-cost compliant gripper using cooperative mini-delta robots for dexterous manipulation. In 17th edition of Robotics: Science and Systems (RSS).*

## 4.1 Introduction

In unstructured settings like hospitals and homes, robots require the ability to execute dexterous manipulation tasks like handling delicate and small objects such as pills and coins. Many existing robotic end-effectors are designed for industrial applications where the focus is on repeatable and robust manipulation of large and rigid objects. However, these end-effectors can exert significant forces that can damage smaller, delicate, and nonrigid objects, like berries and playing cards. Interest in soft manipulators has grown recently because of their advantages in safety and compliance [18]. To leverage these desirable properties in dexterous manipulation, we propose a novel compliant gripper composed of cooperative 3-DOF mini-delta robots that are made using soft 3D-printed materials.

Delta robots are highly effective and accurate for pick and place tasks in a variety of industrial manufacturing and packaging processes [66]. However, utilizing them for other purposes has not been widely studied. In particular, many collaborative robot (cobot) arms used in both academia and industry are outfitted with either two finger parallel jaw grippers or vacuum grippers. Some labs and companies have used significantly expensive and complicated anthropomorphic (human-like) hands such as the shadow hand, which are typically difficult to control autonomously due to their high degrees of freedom [67]. Other researchers have developed their own hands such as soft pneumatic grippers [68] or jamming grippers [69], but they usually have a smaller workspace and less accuracy.

Our gripper, presented in this paper, consists of an end-effector with two 3-DOF delta robot modules as shown in Figure 4.1. In contrast to other grippers, our mini-delta robots use closed-form inverse kinematic solutions and soft materials which achieve high accuracy while still providing compliance. Furthermore, our end-effector is accessible through the use of 3D-printing and readily available off-the-shelf parts. The modular parts can be easily replaced and produced at a low cost. The price for our delta gripper is approximately $300 ($150 per 3-DOF finger),

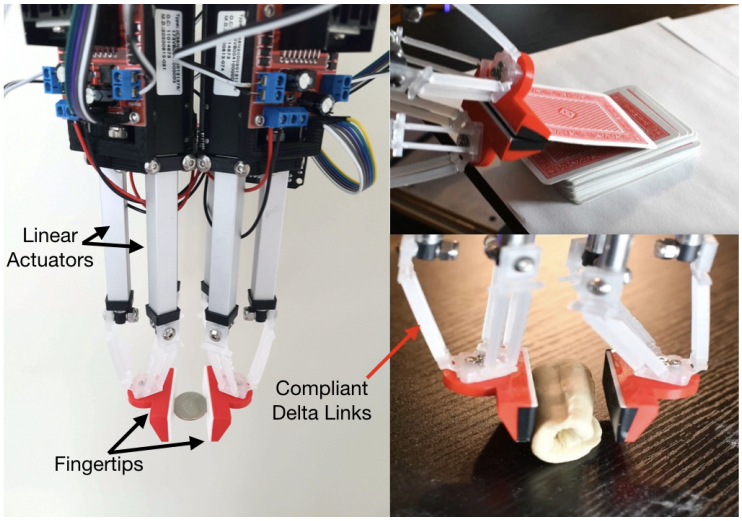



Figure 4.1: Novel robotic gripper grasping a coin, a card, and a dough roll with two delta robots, each with three degrees of freedom and made from 3D-printed soft material, polypropylene (PP). The linear actuators, compliant delta links, and fingertips are shown for a pair of delta robots. The delta robot's links move up and down with the linear actuators fixed at the joints.

which is significantly cheaper than the cost of some off-the-shelf grippers. For instance, the anthropomorphic Shadow Dexterous Hand starts from $50,000 [3] and Robotiq's dexterous 3-finger adaptive robot gripper is $18,000. Even a two-fingered gripper from Robotiq costs around $5,000 [70]. The dexterous dynamixel claw from BAIR lab [49], which also has 3-DOF per finger, costs around $2500 for 3-fingered manipulation. From our total delta gripper cost of $300, the cost of the actuators ($40 each) is the largest cost out of all the delta robot materials. We therefore consider it a low-cost gripper when comparing the magnitude of cost with commercial dexterous grippers. Unlike these grippers, the delta robot gripper with its parallel mechanism is compliant, low-cost, easily manufactured, and modular.

Our main contribution in this paper is the design and modeling of a novel gripper composed of 3-DOF compliant delta robots. We start by presenting a design of the gripper and learn its kinematic model using neural networks as the traditional rigid delta model is inaccurate for compliant links. Subsequently, we construct a force profile of the compliant delta robot in various starting configurations using a one-axis force sensor. Finally, we conclude by using the gripper with the learned kinematic model on several dexterous manipulation tasks including manipulating a grape, aligning a pile of coins, picking up one coin from a pile, picking up a card from a deck, plucking a grape off of a stem, and rolling dough. Through these manipulation demonstrations, we present a multi-fingered hand design that can execute precise and low-inertia manipulations.

## 4.2 Related Work

Delta robots are three translational degrees of freedom (DOF) parallel mechanisms that can be used for manipulation. The advantage of such a robot compared to a serial manipulator is that the inverse kinematics can be computed in closed-form, allowing for fast and easy control. Ad-

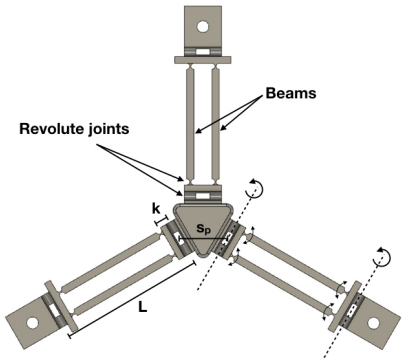



Figure 4.2: This diagram illustrates the two orthogonal axes of rotation that approximate a universal joint and the measurements of $k, s_p$, and $L$. $k$ is shown as the distance between the two orthogonal axes of rotation, $L$ is shown as the distance between the axes of rotation at the top and bottom of a leg, and $s_p$ is shown as the distance between the the axes of rotation where the center of the parallelogram links attach to the end-effector. Each measurement represents a distance constraint between revolute joints in our simulation.

ditionally, the motors are stationed at the base of the delta robot, creating a light end-effector that can move precisely with low inertia. Such low inertia mechanisms combined with compliant materials have a lower chance of harming objects and humans upon interaction. These qualities can be enhanced with the use of soft materials for a safe robotic gripper.

However, soft robots require a departure from classical methods for design, fabrication, and control [71]. The design of our compliant delta robot is similar to the laminate millimeter-scaled delta robot [6] [63] and compliant parallelogram links characterized in [72]. Unique from these two approaches, we use 3D-printing for fabrication and linear actuators to create a prismatic delta rather than rotary motors for a revolute delta. Thus, we use three linear actuators per 3-DOF delta robot to move the end-effectors as shown in Figure 4.1 which behaves practically the same as the revolute delta [73].

Actuation and control of soft robots is an ongoing challenge. Some relevant works have tackled this by using learning-based methods. Truby et. al. [74] use deep learning to map piezoresistive sensor readings to 3D configuration of a complex soft robot. Homeberg et. al. [75] also use proprioceptive sensors but to distinguish objects in-hand. It is common to deduce where the robot is in space through external or internal sensors and in some cases both [76]. In our work, we exploit rigid delta kinematics to build a prior of where the delta end-effector is in space. Then, using a neural network, we create a robust model relating actuator positions to end-effector positions for delta robots of two different soft materials.

In our previous work on 3D-printed compliant delta manipulators [72], we characterized thermoplastic polyurethane (TPU) and polypropylene (PP) parallelogram links for optimal design parameters. The main characteristic of a delta robot mechanism is that the end-effector stays parallel to the base. In our work, we use the optimal dimensions found to maintain this parallel relationship in [72] to design and fabricate a delta robot manipulator using the two materials, TPU and PP.

## 4.3 Delta Manipulator Design

Using 3D-printing and soft materials significantly impacted our design of the delta robot gripper. Our compliant delta links made from PP and TPU require a different design when compared to rigid delta links. The delta design, as shown in Figure 4.2, is also dependent on the gripper workspace and actuators. Additionally, fingertips made from PLA are mounted on the delta robot's end-effector, as shown in Figure 4.1. For comparison, out of the materials used, TPU is the most compliant, followed by PP and then PLA with tensile moduli of 26 MPa (using ASTM D638), 220 MPa (using ISO 527), and 2,346.5 MPa (using ISO 527), respectively [77]. We used an Ultimaker S5 to create all of our 3D-printed parts, but similar fused deposition modeling (FDM) printers are widely accessible, which greatly lowers the barrier of entry to make these low-cost compliant delta robotic manipulators.

### 4.3.1 Setup

In this work, we test delta robots made from two soft materials, PP and TPU, so we design two compliant delta links accordingly. We use compliant parallelogram links with living hinges to 3D-print delta robots as characterized in previous work [72]. The two living hinges rotate along orthogonal axes to approximate a universal joint in the delta links, as shown in Figure 4.2. Each leg is composed of two beams that move as a four bar parallel linkage mechanism, which transfer the motion from the linear actuators to the end-effector, as shown in Figure 4.1. These parallelograms have $0.375$mm hinges, and $2.5$mm and $4.5$mm thick beams for PP and TPU deltas, respectively. We use these values to ensure that the delta robot end-effector remains as parallel to the base, throughout as much of the robot's configuration space, as possible [72].

The delta links are attached to the ends of three ECO LLC Mini Electric Linear Actuators with $76.2$ mm stroke and 20 N maximum load. The actuators are controlled through ROS serial with an Arduino Mega and L298N motor controllers. The total weight of the gripper with two delta modules is 1.03kg. The gripper is mounted on a Franka Panda Robot Arm for additional mobility during task executions. Further dimensions such as the length of the delta parallelogram beams (as shown in Figure 4.2) and distance between the beams are chosen according to the desired workspace of the delta gripper.

### 4.3.2 Delta Actuators and Workspace

The parallelogram links characterized in our previous work [72] were evaluated on revolute delta robots while our delta gripper uses prismatic deltas actuated by linear actuators. While the delta robot mechanism remains the same, the prismatic delta design allows more freedom in the packing of delta robots. Our gripper only features two delta robots as shown in Figure 4.1, although the design framework can be easily extended to more delta robots adjacent to the existing ones. Revolute delta robots would require more space around the rotary actuators, hence using linear actuators instead allows us to pack deltas closer together and enable cooperative capabilities between the robots.

The prismatic delta workspace is generally close to a hemisphere shape as shown in Figure 4.3. The workspace shown only accounts for joint angles until the delta links are perpendicu-

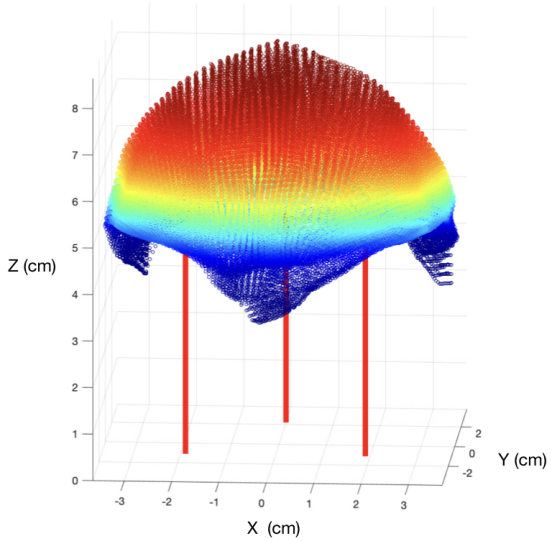



Figure 4.3: The prismatic delta workspace, in centimeters, has a dome-like shape, where the robot can reach any of the points in X, Y, and Z axes. The three vertical red lines represent the position of the linear actuators. Colors are mapped to the height for a better visualization. The actuator lengths are limited to $4$ cm so that no actuator can be above the delta's end-effector. Higher z values can be reached by adding a constant offset to every actuator.

lar to the linear actuators, as significant deformation of the compliant links will occur after this point. We design adjacent delta robots such that they share a section of their workspace to enable cooperative manipulation of an object. Parameters such as the leg length $L$, distance between actuators at the base $s_b$, and distance between where the parallelogram links attach to the end-effector $s_p$ are chosen to create overlapping workspaces between adjacent deltas while taking into account the size of the linear actuators and how close we can pack them. Additionally, we use similar delta structure dimensions characterized in related work [72] except that the parallelogram beams were adjusted to be $6$mm apart from each other and $37$mm in length. These changes allow for larger joint angles and overlapping workspaces.

### 4.3.3 Fingertip design

For our delta gripper, we consider two types of fingertips 3D-printed with $10\%$ infill. First, the planar fingertips made from red Tough PLA shown in Figure 4.1 mimic a parallel jaw gripper in that the contact surfaces are flat and opposable. The second fingertip is spherical in shape and made from blue TPU as shown in Figure 4.7(d), making the compliance at the contact points significantly higher at $10\%$ infill. Additionally, the planar fingertips are padded with $2$mm thick foam and then electrical tape to increase friction and compliance between the fingertip and object. The main compliance of the gripper comes from the soft delta links as the foam padded fingertips are fully compressed after $1$mm of deformation. We exploit the compliance of our delta robots in conjunction with our fingertips to manipulate small and delicate objects dexterously.

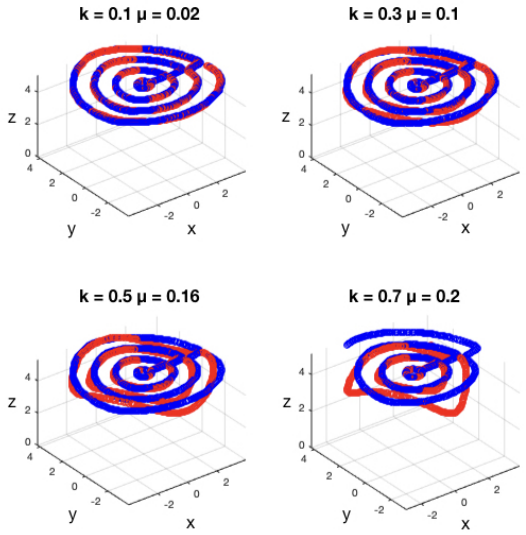



Figure 4.4: Effects of varying the offset $k$ on the mean error $\mu$ from the path of a standard delta. The path of the delta with offsets is shown in red, and the path of a standard delta is shown in blue. For $k = .7$ cm, some actuator inputs become infeasible and the path is cut short.

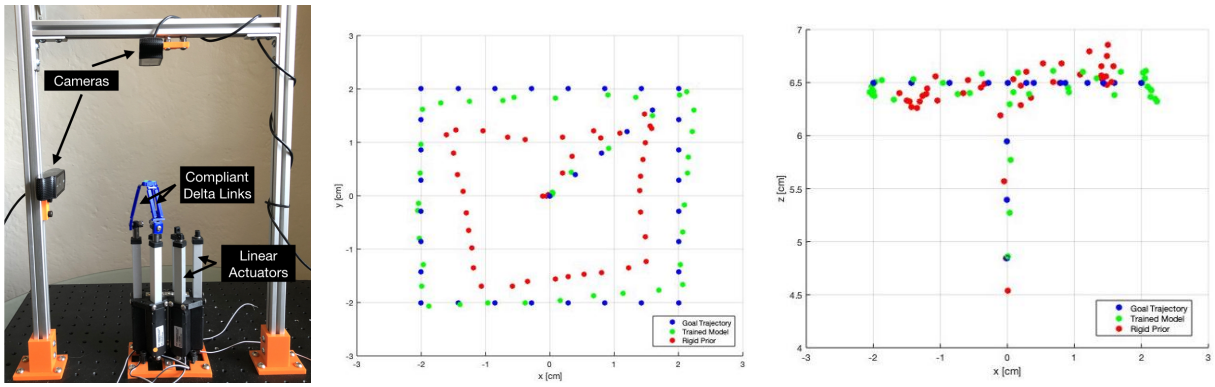
## 4.4 Delta Modeling

There is extensive literature analyzing rigid delta robots [78, 79], but the manufacturing of flexible delta robots through 3D printing introduces significant changes to the kinematics. We approximate each universal joint in the delta with an orthogonal pair of revolute joints that are separated by a small offset $k$ as labelled in Figure 4.2. We analyzed the effect of the offset between revolute joints on the delta workspace by modeling a rigid version of our delta in Simulink. To model the compliant delta kinematics, we use the rigid model as a prior and learn the residual correction using a distal learning approach with data obtained from a marker-based stereo visual tracker.

### 4.4.1 Offset between Revolute Joints

In order to approximate universal joints, our delta has orthogonal revolute joints that are separated by a small offset. We studied the effect of these offsets on the workspace by simulating a rigid version of our flexible delta robot. We measured four values from our delta robot to parameterize the simulation: leg length $L = 4.8$ cm, distance between linear actuators $s_b = 4.3$ cm, distance between where the legs attach to the end-effector $s_p = 1.6$ cm, and an offset $k = .5$ cm (see Figure 4.2).

In Figure 4.4, we studied the effect of varying $k$ by measuring the deviation from the kinematics of a standard delta robot on a test trajectory. The test trajectory is a spiral that moves from the center of the workspace to its edge in the XY plane. The z coordinate is selected to maximize the width of the workspace. This path is then representative of the workspace as a whole because all points can be reached at other values of z by adding a constant offset to each of the three linear actuators. Figure 4.4 shows that for a large $k$, some actuator inputs become infeasible




(a) (b) (c)

Figure 4.5: (a) Marker-based stereo perception system to track the delta end-effector position using two orthogonal Logitech C920 cameras. The graphs compare measured and desired delta positions on the (b) XY and (c) XZ plane when following a test path (shown in blue) with the PP delta. The neural network trajectory prior to training is shown in red, and the trajectory after training is shown in green.

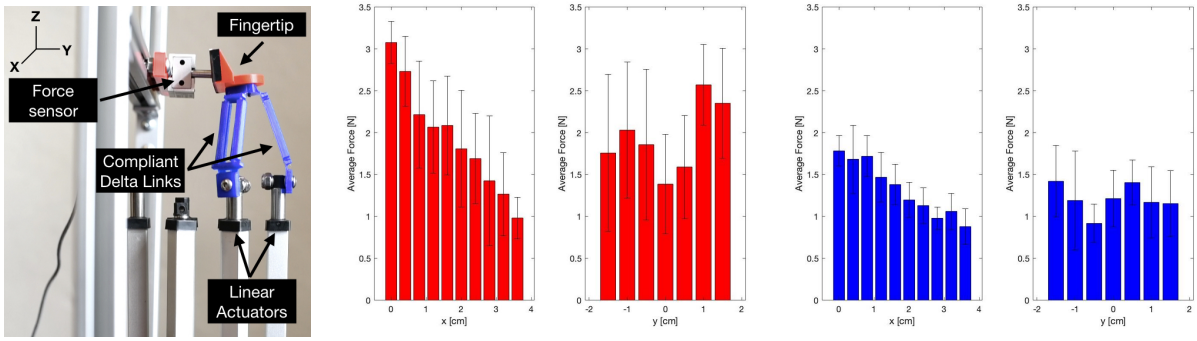
when the delta is near the edge of the workspace. Our value of $k = 0.5$ warps the standard delta workspace by an average of $.16$ cm on our test path, and does not cause any actuator inputs to become infeasible.

The most significant source of error when compared to the standard rigid delta is the deformation and twisting of the links. Each revolute joint in the delta applies a torque towards its rest position when actuated. In a rigid delta, this force is counteracted by the actuators, but for flexible deltas, equilibrium can be reached by changing the shape of the robot itself. Therefore, deformation occurs as joint angles become large. There are multiple ways that the robot is able to deform, and the type of deformation that occurs is determined by the robot's design parameters. Since we chose a high ratio of beam link width to joint hinge width in the parallelograms, the links themselves do not bend significantly. Instead, the revolute joints are able to twist a small amount, which can lead to large changes in the position of the end-effector.

### 4.4.2 Learning Delta Robot Kinematics

We used a marker-based stereo perception system to track the position of the real world delta and collect data to learn the forward and inverse kinematic models for the flexible delta robots as shown in Figure 4.5. To acquire the accurate kinematics, we trained a neural network that was pretrained to match the kinematics for our rigid delta model. We trained two different neural networks to model forward and inverse kinematics. Each network has $3$ densely connected ReLU layers with $256$ hidden units each and linear activation at the output.

The networks were trained using a distal teacher approach [80] where the forward kinematic model was trained to match input actuator positions with measured end-effector positions, and the inverse kinematics model learned inputs to the forward network that would reduce the error between its prediction and the measured position of the robot. This structure ensures consis-




(a) (b) (c)

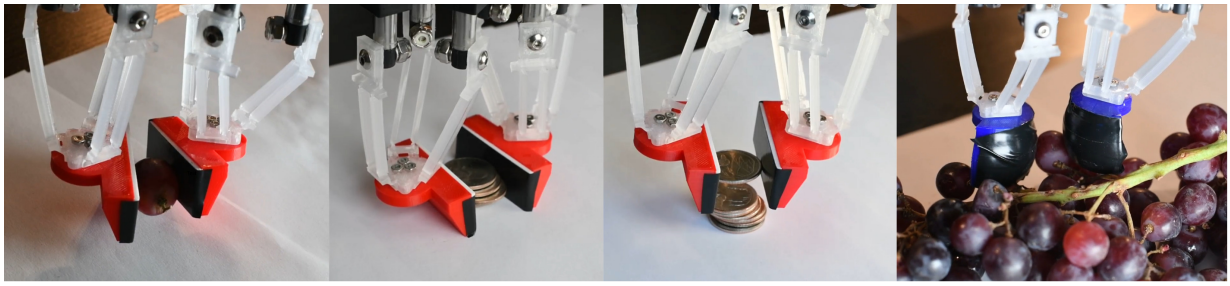
Figure 4.6: (a) Force profile experiment setup consisting of a GSO-500 Transducer Techniques Load Cell and TPU delta robot with planar fingertip. The delta robot pushes on the load cell with a displacement of $5$mm at various positions in the workspace, along X and Y axes. The mean force exerted by the delta at different values of x and y, and standard deviation are shown for the (b) PP deltas (c) TPU deltas.

tency between the network outputs, and it makes it easy to identify and target areas where the kinematics are not well-known because the two networks will produce conflicting results. The training data was also augmented using the symmetry of the delta robot. Observed end-effector positions were copied and reflected over the y-axis, corresponding to switching the heights of the two rightmost actuators in Figure 4.5. The actuator and end-effector positions for each observed point were also rotated by $\pm 120$ degrees about the z-axis. Finally, each observed end-effector position was given multiple z offsets, corresponding to adding a constant to the height of every actuator. These changes ensured that the learned workspace would be symmetrical.

Separate models were trained for both the TPU and PP deltas. The pretrained rigid delta model network had a mean error of $1.3$cm and $0.72$cm from the test path for TPU and PP deltas, respectively. To improve our pretrained network, it was sufficient to teach the network the flexible delta kinematics by running 100 trajectories that took approximately 20 seconds each. The models were then evaluated based on accuracy and repeatability when following a test path 50 times, as shown for PP deltas in Figure 4.5(b)-(c). After training on the TPU delta data, the mean error from the goal path was $0.33$ cm, and the mean pairwise error over 50 trajectories was $0.13$ cm. The mean error from the goal path for the PP delta was $0.28$ cm, and the mean pairwise error over 50 trajectories was $0.09$ cm. By fitting models to each type of delta, we were able to decrease the kinematics model error significantly for both deltas and confidently deployed them during our robot experiments.

## 4.5 Force Profile

As discussed in Section 4.4, deformation of the delta links occurs as joint angles become large, which happens towards the edges of the workspace. To determine whether this effect weakens the payload capacity of the delta in certain configurations, we displace the end-effector by a fixed



(a) (b) (c) (d)

Figure 4.7: Panels (a), (b), and (c) show the delta grippers with planar fingertips 3D-printed using PLA grasping a grape, aligning a pile of coins, and taking a coin from that pile and rotating it in hand, respectively. Panel (d) shows the deltas with spherical TPU fingertips picking a grape from its stem.

distance along X and Y axes, and measure the resulting force to create a force profile of the delta.

For a given end-effector position $(X, Y, Z_1)$, another end-effector position $(X, Y, Z_2)$ may be achieved by offsetting all three of the linear actuators, as shown in Figure 4.6(a), by $Z_2 - Z_1$. We selected the $Z$ value that maximized the width of the delta workspace and calculated the force exerted by the delta gripper on the resulting XY plane. Any force measurement at a point $(X, Y, Z_1)$ is representative of the delta's force output at any other point $(X, Y, Z_2)$. We sampled the x-axis in 4mm increments and the y-axis in 5mm increments from the point $x = -1.5$cm, $y = -3.6$cm to the point $x = 1.5$cm, $y = 3.6$cm. Accounting for symmetry, only positive x-axis values are taken into consideration for measurements.

To test the force at a certain position of the workspace, the delta robot end-effector with a planar fingertip was moved to contact the load stem of a GSO-500 Transducer Techniques Load Cell. Then, we recorded the blocked force exerted by pushing the delta at the center of the fingertip surface $1$, $2$, $3$, $4$, and $5$mm into the load cell, as shown in Figure 4.6(a). Testing both TPU and PP delta robots, we observed that the force exerted grows linearly with the increased displacement. To measure the linearity, we calculated the $R^2$ value after linear regression for the five force measurements at each coordinate. On the TPU delta, the mean $R^2$ value across all measured points was $0.9541$, with a standard deviation of $0.1183$. The mean $R^2$ value for the PP delta was $0.9730$ with a standard deviation of $0.0393$. This linear relationship allows us to control the force exerted by the delta robot through its displacement.

We grouped the data based on the x and y coordinates, and reported the mean blocked force when displacing the delta $5$mm in the direction of the load cell. Figure 4.6(b)-(c) shows that increasing the value of x (moving parallel to the plane of the fingertip away from the center of the workspace) decreases the force output of the delta end-effector. There is no clear trend between the y coordinate of the delta and the mean force that can be exerted. This may be due to the orientation of the delta end-effector changing as it moves forward or backwards along the y-axis.

## 4.6 Experiments

As a result of our work in Section 4.4, we are able to execute delta manipulator trajectories with precision. To further test the manipulator, we evaluate the success of manipulating various small objects with open loop control or human teleoperation using a PS4 Dualshock Controller. The six tasks we executed are as follows, 1) in-hand manipulation of a single grape, 2) aligning a pile of coins, 3) picking up a coin and rotating it in-hand, 4) slide-to-grasping a card from a deck, 5) twisting a grape off of its stem, and 6) rolling up dough between the fingers on a table. Unlike rigid manipulators, our soft delta gripper can exploit contacts, similar to a human hand, to execute tasks precisely. We chose these tasks to demonstrate the compliance of the deltas and their ability to manipulate delicate objects. While existing grippers may be able to execute these tasks using additional DOFs, we present a unique gripper that can perform all six tasks as a proof of concept. Due to our force profile experiments in Section 4.5, we used the PP delta in all of our demos due to the higher force it can exert. All of the demos are shown in the accompanying supplementary video[1].

In Figure 4.7(a), we show the delta gripper using planar fingertips to grasp a grape. Even when the delta robots use their maximum force to squish the grape, the compliance in the deltas prevent it from being crushed. Instead, the delta fingertips twist, while still holding on to the grape.

Next, in Figure 4.7(b), the deltas arrange a pile of coins by executing two parallel grasps that are orthogonal to each other. While this demo could also be completed by a parallel jaw gripper, there is a chance that the coins would fly out of the gripper if too much force was exerted on the pile. Our deltas gently align the pile of coins in order to create the precisely aligned pile. After the gripper aligns the coins in the pile, it grasps the top coin as in Figure 4.7(c) and is able to rotate the coin in hand. This task illustrates the ability of our robot to move in an additional axis that normal parallel jaw grippers cannot.

Using spherical TPU fingertips the delta gripper picks a grape off of a stem in Figure 4.7(d). Taking a grape off a stem requires the robot to twist the grape in order to apply the necessary pressure on the stem to get it to release without damaging the grape. The spherical fingertips allowed the grape to roll in between the fingertips, resulting in a twisting motion. Afterwards, the robot was able to remove the grape from the stem. This motion required human teleoperation as it involved positioning the fingertips so as to not allow other neighboring grapes to impede the motion of the deltas.

The final two tasks are illustrated in Figure 4.8. In the card pickup task, the top delta robot uses a stroking motion in order to slide the top card from the rest of the deck. Afterwards, the bottom delta lifts up and pinches the card together with the top delta to pick up a single card. The slide-to-grasp motion is made possible by the gripper's additional degrees of freedom and compliance. The sliding motion was programmed to execute autonomously, although it heavily depends on the initial positioning and orientation of the deltas relative to the cards.

The last task involved rolling a flat piece of dough into a spiral roll. This task also required human teleoperation due to the inherent compliance of the dough itself. One fingertip was used to mainly hold the dough in-place while the other was executing a scooping motion in order to

[1] https://youtu.be/yciJn3rgFHw

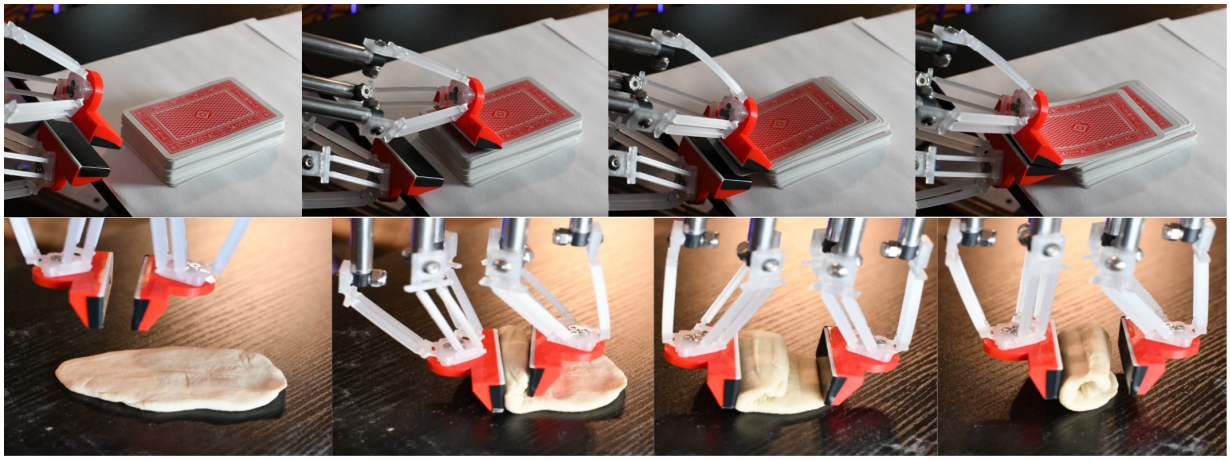


Figure 4.8: Timelapse of the compliant delta gripper sliding the top card and picking it up from the deck, and rolling a flat piece of dough into a spiral.

get under the dough and push it. Without the degrees of freedom provided by the deltas, this task would likely be difficult for most grippers.

Throughout all of the demos, the compliance of the deltas and added degrees of freedom enabled a wider range of motion that normal parallel jaw grippers would not afford. In addition, as we had the inverse kinematics for the delta robots, we were able to quickly translate a desired trajectory into commands to the linear actuators. This direct mapping allowed us to easily teleoperate the robot with a PS4 Controller to complete tasks that would typically require a motion tracking hand setup in order to give the robot demonstrations [81]. In the future, we plan to explore more delicate and dexterous tasks with added sensors to provide feedback when interacting with objects.

## 4.7 Discussion and Conclusion

Through kinematic modeling, force profile characterization, and manipulation task executions, we explored the capabilities of compliant delta grippers made from two soft materials, TPU and PP. While the two materials vary significantly in compliance, the learned kinematic models for the TPU and PP deltas did not differ significantly in performance. Additionally, the force profiles were similar in their ability to exert maximum forces at the center of the workspace. We expect similar trends to extend to delta robots made from materials similar to TPU and PP.

Our kinematic model learning error and force profile experiments show that the delta gripper is easy to control. The robot experiments show that dexterous manipulation tasks such as rolling dough and picking a grape off its stem can be executed with the degrees of freedom provided by each delta. Additionally, the compliance of the delta robot avoids damaging items like the grape. Thus, we can ensure that the delta robot can manipulate delicate objects and interact with its environment safely.

We present the groundwork for creating multi-fingered hands that can execute precise and low-inertia manipulations. Future extensions of this work can explore grasp planning and in-

creasing the number of cooperative deltas to handle larger objects. In addition, we plan on incorporating internal and external sensors to the deltas in order to use visual and haptic feedback for more precise autonomous manipulation.

However, a limitation of our two finger gripper using delta robots was that manipulations were restricted to two contact points. Many dexterous tasks require contact-rich manipulation, which motivates our next work involving four-finger anthropomorphic hands to tackle a wider variety of dexterous tasks.

# Chapter 5

# Designing and Evaluating Tendon-Driven Soft Hands

**Paper:** *Mannam, P., Shaw, K., Bauer, D., Oh, J., Pathak, D., & Pollard, N. (2023). Designing anthropomorphic soft hands through interaction. In IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids).*

## 5.1 Introduction

Rapid prototyping technologies have advanced significantly, making way for designers to build new systems at a fast pace. These techniques, such as 3D-printing, allow for quick turnaround between design iterations to test and evaluate systems quickly. This is especially useful for systems with dynamics that are difficult to predict or model, such as soft robot manipulators.

Iterating for dexterous soft hand designs is a laborious process. The complex design space and the infinite degrees of freedom make it difficult to predict the effects of incremental design changes. Unlike rigid robot hands, state-of-the-art soft body simulators are not able to provide accurate, efficient, and robust evaluation of soft designs [82]. Hands such as the BRL/Pisa/IIT SoftHand [83] or the RBO hand [35] have evolved over years to incorporate more adaptive synergies and dexterity. To speed up development times and reduce fabrication overhead many works have recently turned towards 3D-printing to either directly print soft hands [7] or to quickly create complex molds [84]. While this has significantly reduced the cycle time for fabrication, designing dexterous soft hands still requires a lot of expertise, and trial and error due to the continuously deformable nature of soft robots. The lack of appropriate simulators means the evaluation of soft hand designs has to be done on the real prototype by using hand-crafted policies [85] or sequential keyframed open-loop poses [7].

Our key insight is that we can evaluate these systems beyond hand-crafted policies or sequential keyframed open-loop poses using recent advancements in teleoperation systems. Improvements in hand tracking and pose estimation [86] have led to the development of vision-based teleoperation approaches including using a single RGB camera for real-time tracking of human hand poses [87]. Teleoperation offers valuable insights into system performance and enables the identification of robust strategies in real-world scenarios. Simulation often falls short in cap-

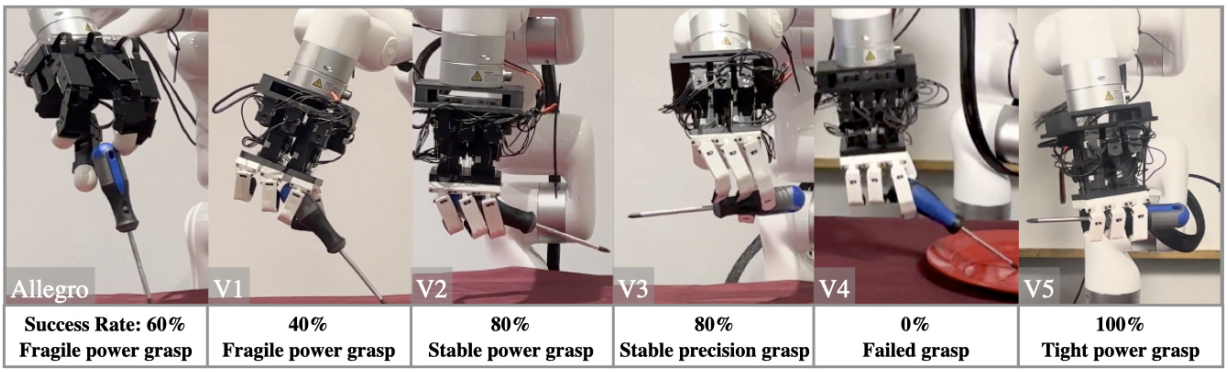



Figure 5.1: Manipulation task performance over five iterations of DASH designed through rapid prototyping and real-world evaluation on tasks alongside task performance of our baseline hand Allegro.

turing system nuances, accurately modeling soft materials, and adapting strategies. Therefore, tasks that succeed in simulation can still fail when observed and evaluated through teleoperation, providing a clearer understanding of the system's capabilities and limitations.

In this paper, we 3D-print soft robotic hands to test iteratively using teleoperation on a designated manipulation task set, modify the design, and repeat this process shown in Figure 5.2(a). While manually testing and revising designs to improve systems is not a new concept [84], recent technologies allow us to repeatedly iterate the entire framework of designing, fabricating, and evaluating in a matter of *days*. This framework is versatile and can be implemented at any stage of the design process. Its purpose is to bridge the gap between the real world and simulation by adjusting for discrepancies or expediting fine-tuning for real-world scenarios. We envision this framework as a valuable augmentation to existing design frameworks, enhancing the overall design process for soft anthropomorphic robotic hands.

Using our framework, we present a case study to design a 16-DoF tendon-driven 3D-printed soft hand DASH, shown in Figure 5.1. This hand has a small form factor similar in size to a human hand, 3D-printable parts that are easily replaceable, and a modular customizable design that allows for easy iteration. Through teleoperation, we explore the capabilities of the soft hand in order to inform our design iterations across five hands: DASH v1, v2, v3, v4, and v5. In order to evaluate the dexterity of the hand, we designed a suite of 30 manipulation tasks with varying grasp types and objects that are inspired by human hand capabilities, which allows us to test the capabilities of our robot hand. Our hands show improvements across iterations, albeit not monotonically, and each iteration, except v1, required less than 100 hours to design, fabricate, and test. DASH v1, v2, v3, v4, and v5 succeed on 70%, 82%, 83%, 75%, and 87% of executions across all tasks, respectively. We also outperform a commercial dexterous robotic hand, Allegro [2], which has a success rate of 60% on the same 30 tasks.

The contributions of this paper are

- A detailed report of our process for designing soft hands that leverages rapid prototyping techniques and uses a teleoperated real robot for evaluation, instead of simulation.
- The design of a state-of-the-art dexterous anthropomorphic soft hand using our framework, that outperforms a commercial robotic hand on real world manipulation tasks.

- The release of open-source CAD models and data corresponding to 900 teleoperated human demonstrations to democratize access to low-cost dexterous hands.

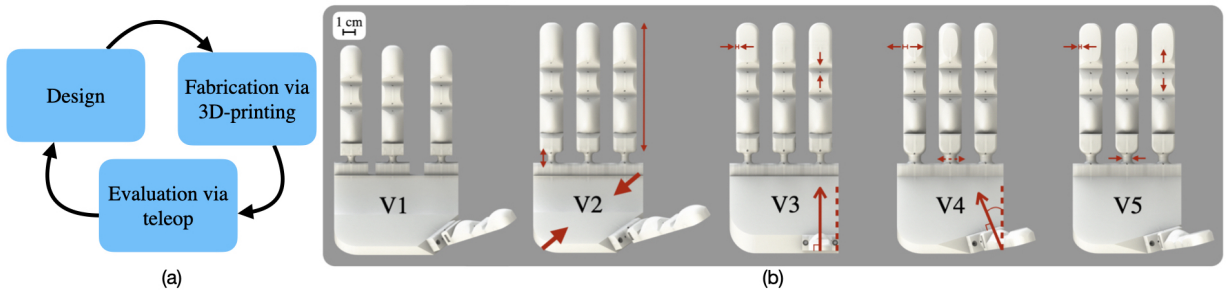



Figure 5.2: (a) Our soft robotic hand design process involving rapid prototyping and real-world evaluation (b) CAD models and differences across DASH iterations V1 through V5, as explained in Section 5.4.

## 5.2 Related Work

Soft robotic hands such as RBO [35] utilize intrinsic compliance, rapid prototyping, and actuated palms for a modular, highly compliant, high degree-of-freedom, and low cost manipulator in order to perform a large variety of in-hand manipulation tasks and object grasps. However, existing robotic hands are still far from achieving human-like dexterity for manipulation [88]. It is necessary to continuously improve and refine the design of both existing and new robotic hands in order to achieve greater dexterity and functionality for performing more complex manipulation tasks.

Although there is currently no unified framework for designing iterations of soft anthropomorphic hands, design iteration methods for robotic systems have been explored. Typically, Finite Element Method (FEM) is used to assess optimal geometries and morphologies before fabricating the final design for real-world evaluation [89, 90]. For instance, the SOFA soft robot simulator has been used to co-optimize control and design of soft hands [43]. However, these approaches are primarily limited to simulated environments and do not address the sim-to-real gap or real-world design iteration. A related framework in robotic fish design utilizes simulation-based FEM testing, real-world design iteration, and proposes a modular design for easier iterations [84]. Nevertheless, optimizing soft robots is challenging due to their complex geometries, impeding the development of efficient optimization algorithms. Furthermore, the lack of efficient simulation tools that can rapidly evaluate design candidates further compounds the challenge [82].

Learning control policies for dexterous manipulation is challenging due to the high DoFs and complex interactions [67, 91]. In contrast, teleoperation offers a swift and natural way to control robot hands, beyond pick-and-place scenarios. It has been used for human demonstrations in imitation learning [92, 93]. Teleoperation is particularly valuable during the design process of dexterous hands, enabling quick evaluation of nuanced capabilities. Mapping human to robot hand morphology can be categorized as *Joint-to-Joint*, *Point-to-Point*, or *Pose-based* [94]. For our soft hand, we adopt a similar joint-to-joint mapping technique as Liarokapis et al. [95].

Our work extends the design and fabrication methodology presented by Bauer et al. [7], consisting of simplifying the design complexity of soft hands by incorporating geometric features such as bumps or creases to achieve 'joint-like' deformations. They perform kinematic testing on designs before fabricating and evaluate a single design. Similar to Bauer et al. [7], we utilize 3D printing, creases for 'joint-like' deformations, and tendon-driven actuation to curl the soft fingers. In addition to these features, DASH's design incorporates three additional tendons in all fingers, enabling adduction, abduction, and folding of the fingers towards the palm, thereby enhancing dexterity.

## 5.3 Experiment Setup

### 5.3.1 Robot Hand design

#### Finger Joints

All iterations of DASH consist of four fingers: the thumb, index, middle, and ring fingers (see Figure 5.3(a)). In order to achieve modularity, each finger, including the thumb, is designed identically. Each finger has three joints (from the base of the finger to the fingertip): the metacarpophalangeal (MCP) joint, proximal interphalangeal (PIP) joint, and the distal interphalangeal (DIP) joint. The joints for each finger are shown in Figure 5.3(b).

#### Tendons

Each finger is controlled by four tendons shown in Figure 5.3(b). Two tendons run along the sides of the MCP joint, closest to the palm, for abduction and adduction, which allows the fingers to move closer together and farther apart. These two tendons are controlled by a single motor, so we refer to them as tendon 0. A single tendon, tendon 1, is used to flex the finger forward at the MCP joint, orthogonally to the axis of motion of the abduction-adduction tendons. The last tendon, tendon 2, runs through the entire length of the finger to enable completely curling into itself.

### 5.3.2 Fabrication using 3D-printing

The hand assembly, shown in Figure 5.3(a), is the same for all hand iterations and consists of 4 soft fingers attached to the soft palm. The rigid components include a top plate below the palm, 12 motors, a bottom plate which also houses the Dynamixel U2D2 motor controller, and a xArm6 mount.

The soft dexterous hand's mounts, motor housing, and motor pulleys are all 3D-printed from PLA (rigid material depicted in black in Figure 5.3(a)) while the soft hand was printed with Ninjaflex Edge (83A shore hardness) [96] using an Ender 3 S1 Plus. For all the robot experiments, the hand was mounted onto a xArm6 [97] robot arm. DASH costs approximately $\$1500$ to build with the majority of the cost consisting of 3D-printer ($\$500$) and the twelve motors ($\$1000$) required. For comparison, the Allegro hand is also 16-DoF and costs around $\$15000$ [49].

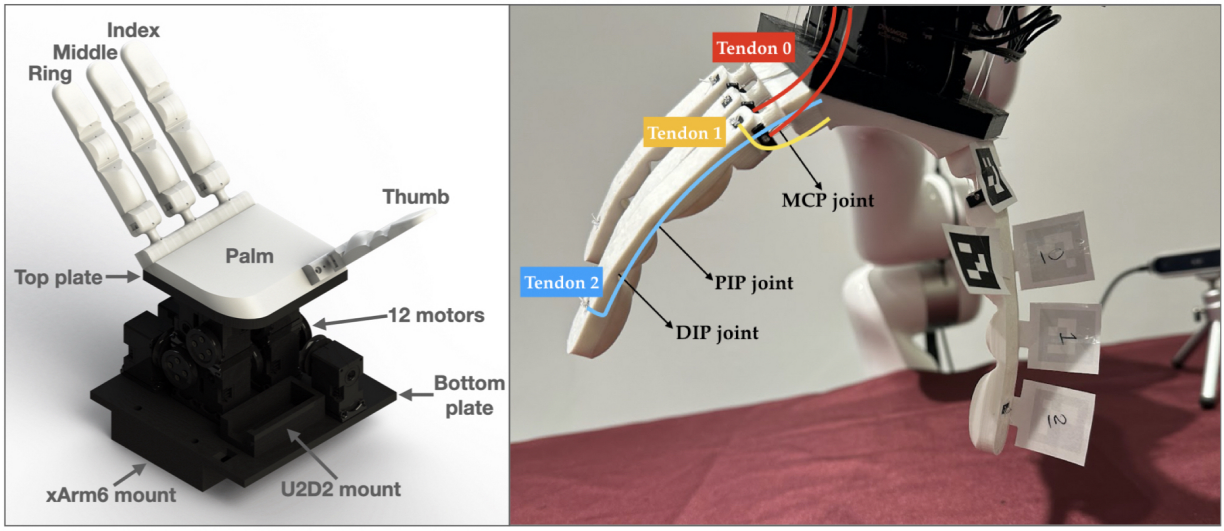



Figure 5.3: (a) Assembly of DASH-v3 (top to bottom) including fingers, palm, top plate, motors, bottom plate, xArm6 mount. (b) Calibration procedure to map motor angles to finger joint positions, where tendon 0 actuates MCP side-to-side, tendon 1 actuates MCP forward folding motion, and tendon 2 curls the finger controlling both PIP and DIP joints.

### 5.3.3 Hand Evaluation using Teleoperation

#### Learning Kinematics

To approximate the kinematics of the finger joints without real-time feedback, we learn a model from collecting offline paired motor and joint angle data from a single finger. Through small increments of 3 degrees of actuation and joint angle tracking across 1000 finger configurations using AR tags and RGB cameras, we obtain a collection of tuples (joint angles, motor angles) that are normalized to $[0, 1]$ for independent kinematic calibration, assuming fixed joint lengths and bending at the creases.

A linear model is learned using the collected data to map finger joint angles to motor angle outputs. We refer to the motors controlling tendons 0, 1, and 2, as shown in Figure 5.3(b), as motors 0, 1, and 2, respectively. The equations for MCP, PIP, and DIP joint angles are shown below. In equation 5.1, we learn the MCP joint angles $\theta_{\text{MCP}_{\text{side}}}, \theta_{\text{MCP}_{\text{fwd}}}$ jointly since the amount of side-to-side angle at the MCP joint can restrict the forward folding motion of the finger. In equation 5.2, the Motor 2 angle $\theta_{\text{motor}_2}$ is an average measure of the motor angle for the desired PIP and DIP joint angles $\theta_{\text{PIP}}, \theta_{\text{DIP}}$ since the same tendon controls both the PIP and DIP joints. The weights in equations 5.1 and 5.2 are found by fitting our data using linear functions. We collect training data for almost two hours for each iteration of the hand to calibrate new models, using the weights shown in Table 5.1.

$$\begin{bmatrix} \theta_{\text{motor}_0} \\ \theta_{\text{motor}_1} \end{bmatrix} = \begin{bmatrix} w_1 & w_3 \\ w_2 & w_4 \end{bmatrix} \cdot \begin{bmatrix} \theta_{\text{MCP}_{\text{side}}} \\ \theta_{\text{MCP}_{\text{fwd}}} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{5.1}$$

| ***Hand Design*** | *v1* | *v2* | *v3* | *v4* | *v5* |
|---|---|---|---|---|---|
| $w_1$ | $-1.05$ | $-0.43$ | $-0.43$ | $-0.59$ | $-0.59$ |
| $w_2$ | $0.01$ | $0.2$ | $0.2$ | $-0.12$ | $-0.19$ |
| $w_3$ | $0.1$ | $0.51$ | $0.51$ | $0.26$ | $-0.32$ |
| $w_4$ | $0.83$ | $0.54$ | $0.54$ | $0.38$ | $0.72$ |
| $w_5$ | $0.67$ | $0.6$ | $0.6$ | $0.62$ | $0.63$ |
| $w_6$ | $0.99$ | $0.76$ | $0.76$ | $1.69$ | $0.65$ |
| $b_1$ | $0.47$ | $0.38$ | $0.38$ | $0.45$ | $0.58$ |
| $b_2$ | $-0.07$ | $0.01$ | $0.01$ | $0.44$ | $-0.03$ |
| $b_3$ | $0.03$ | $-0.04$ | $-0.04$ | $-0.05$ | $-0.09$ |
| $b_4$ | $-0.01$ | $-0.16$ | $-0.16$ | $-0.3$ | $-0.07$ |

Table 5.1: Calibration weights for all five iterations of DASH mapping from finger joint angles to motor angles.

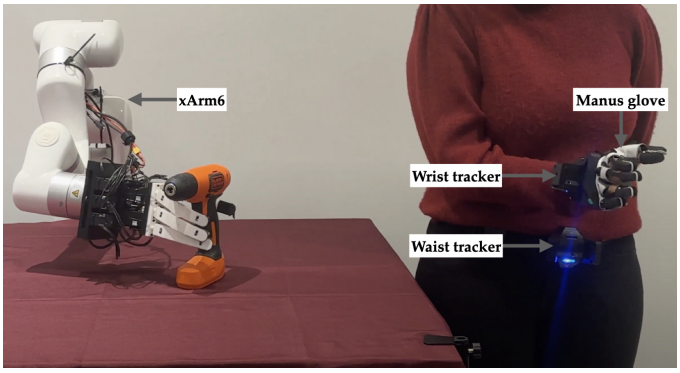



Figure 5.4: Manus Meta Quantum Metagloves used for tracking the hand for teleoperating the robot arm and DASH.

$$\theta_{\text{motor}_2} = \frac{\theta_{\text{PIP}} w_5 + b_3}{2} + \frac{\theta_{\text{DIP}} w_6 + b_4}{2} \tag{5.2}$$

### Teleoperation System

We use Manus Meta Quantum Metagloves [98] designed for VR tracking and Mocap Use, as shown in Figure 5.4 (costs $\sim$ \$8000). The Manus glove is worn on the operator's hand and tracks fingertip positions within a $0.1$-degree accuracy using hall effect sensors. Each finger returns $4$ angles $\theta_{\text{MCP}_{\text{side}}}, \theta_{\text{MCP}_{\text{fwd}}}, \theta_{\text{PIP}}, \theta_{\text{DIP}}$ in real time, which are mapped one-to-one to the robot hand. Then, we convert to motor angles using our kinematics models.

To control the robot arm shown in Figure 5.4, we employ wearable SteamVR trackers [99], utilizing time-of-flight lasers emitted from SteamVR Lighthouses positioned around and above the operator. One tracker is worn on the glove, while another is placed around the waist. We align the waist tracker's rotation with the robot's base frame and adjust the end-effector pose to match the orientation of the human wrist. Then, the human wrist poses are scaled up to cover the robot's larger workspace, making necessary adjustments to ensure user comfort. Safety checks,

including dynamic force feedback on the arm, prevent accidental damage to the robot or its surrounding environment.

### 5.3.4 Manipulation Tasks

Each DASH iteration is tested on a suite of tasks, named DASH-30, listed in Table 5.2 which were inspired by the different types of grasps defined in Liu et al. [21] and tasks from previous teleoperation works [87, 100]. These tasks are categorized by the type of grasp or force necessary. Categories like Hold include a greater number of tasks aimed at testing different grasping techniques and objects. On the other hand, skills like Lever or Twist involve fewer tasks specifically designed to assess whether a particular hand design can successfully perform these skills. Additionally, some tasks were hand-picked as tasks where compliance of the hand may be advantageous.

The feedback from the manipulation task evaluation combines observations from the following metrics: task success, performance across five repetitions of the same task, trends in tasks the hand fails to complete, type of grasps possible or used, opposability of fingers, and reachability of fingertips.

| |
|---|
| ***Hold*** |
| 1) Scissor, 2) Hammer, 3) Chopsticks (single), 4) Pen,<br>5) Wooden cylinder (using adduction/abduction),<br>6) Screwdriver, 7) Drill, 8) (Plastic) Egg*,<br>9) (Plastic) Chip*, 10) M&M* |
| ***Pick (and place)*** |
| 11) Dry-Erase Board Eraser, 12) Tennis Ball,<br>13) Softball, 14) Cloth*, 15) Plush Broccoli,<br>16) Plush Dinosaur,17) Pringles Can, 18) Spam Box,<br>19) Mustard Bottle, 20) Wine Glass, 21) Bin picking |
| ***Lever*** |
| 22) Cube flip, 23) Card pickup from deck |
| ***Twist*** |
| 24) Dice rotation in-hand, 25) Grape off of stem* |
| ***Open*** |
| 26) Plastic bag*, 27) Drawer |
| ***Put in/on*** |
| 28) Cup Pouring (onto plate),<br>29) Cup Stacking & unstacking, 30) 1 inch Block stacking |

Table 5.2: DASH-30: task set of 30 manipulation experiments. Tasks with the asterisk (*) were hand-picked as tasks where compliance of the hand may be advantageous.

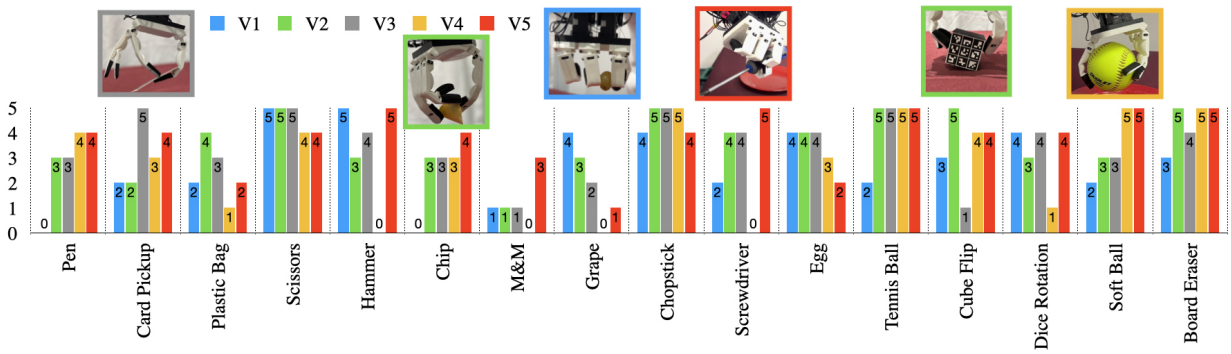



Figure 5.5: Subset of tasks with different performance success across V1 to V5 on specific tasks used to inform design iteration. The top row of inset images shows representative tasks of successful tasks for each hand.

## 5.4 DASH Iterative Design Studies

### 5.4.1 Iteration V1

DASH-V1 performs best on pick and place tasks and tasks using adduction-abduction such as the grape task, shown in Figure 5.5 due to the larger hand size and space between fingers as illustrated in Figure 5.2(b).

#### Design & Fabrication

We start by designing the shape of the finger to enable curling fully into itself, incorporating four total tendons per finger, and redesigning the MCP joint as a multi-axis flexure for increased dexterity in our desired hand. We iterated on the finger design for approximately four months and use this design for all four fingers in DASH-V1.

Designing V1 included considerations such as tendon anchors, 3D-printing settings, and material stiffness. For example, printing the hand with more infill makes it stiffer but requires more torque than our motors can supply for the joint's full range of motion. To better understand the stiffness of the fingers, we test the finger strength by curling the finger completely and using a force gauge to pull on the finger until it uncurls (see Table 5.3 for results). DASH-V1 hand design is shown on the left in Figure 5.2(b). We designed the full hand assembly for V1 in 1 month.

#### Evaluation

We test DASH-V1 on the 30 manipulation tasks from Section 5.3.4, repeating each task five times. Over 150 repetitions, DASH-V1 succeeded on all 5 repetitions for 10 of the 30 tasks, as shown in Figure 5.6. For V1, these tasks were Scissors, Hammer, Wooden cylinder, Cloth, Plush Broccoli, Plush Dinosaur, Pringles can, Mustard bottle, Wine glass, and Cup stacking. V1 struggled to grasp small objects such as the M&M, Pen, and Chip since the fingers were not able to reach and properly oppose each other. For tasks involving precise motions such as picking the

Grape off a stem and opening the Plastic bag, V1 uses the abduction-adduction capability. The abduction-adduction grip strength of V1 is high and enables picking up objects such as the grape off the stem with ease, as shown in Figure 5.5 inset.

V1 succeeds on five out of five repetitions on 10 tasks but shows room for improvement. Grasps that require all four fingers such as picking up a tennis ball would be more successful if the thumb could reach and oppose the rest of the fingers. The best opposability to the thumb was to the ring finger, hence pinch (or precision) grasps were easiest to execute with those two fingers. Improving the reachability and opposability of the fingertips requires a smaller palm or longer fingers. We explore these design options in V2 in order to have more overlap in the workspace of the fingers.

### 5.4.2 Iteration V2

DASH-V2 performance improves on pick and place and hold tasks requiring power grasps. Furthermore, V2 excels at the levering task of cube flip on table, shown in Figure 5.5 inset, due to higher finger strength and fingertip reachability from a smaller palm and longer finger hand design shown in Table 5.3 and Figure 5.2(b).

#### Design & Fabrication

The second iteration of DASH consists of changes to the size of the hand and the MCP joint of the finger. To allow for more reachability among the fingers as well as opposability, the fingers were made longer and the palm was made smaller as shown in Figure 5.2(b). For comparison, DASH-V2 is similar in size to the average male hand which is 88.9mm wide and 193mm long (wrist to fingertip) [5] . Compared to DASH-V1, there is more than a 25% reduction in area of the palm and the finger length increased by 11% in V2 which is shown in Figure 5.2(b).

The MCP joint was improved to achieve a larger range of motion. The underlying structure of the MCP joint is a cylinder to act as a multi-axis flexure, thus we increase the height of the cylinder to increase the joint angle range for the side-to-side and forward motion of the fingers. The design changes also resulted in a higher maximum load of a single finger as shown under finger strength in Table 5.3. Thus, V2 achieves increased range of side-to-side and forward motion for the fingers by redesigning the MCP joint, and has a larger overlap in the workspace of the fingers solving the reachability and opposability issues in V1.

Designing, printing, and assembling V2 took 5, 83, and 6.5 hours, respectively. Printing V2 required us to not only re-print the soft hand, but also the rigid motor housing as the motor arrangement differs from V1. In total, making V2 from V1 took 94 hours.

#### Evaluation

With larger range of motion at the MCP joint and better reachability, we expect V2 to achieve better performance on tasks involving smaller objects like M&M, Pen, and Chip. As shown in Figure 5.5, V2 did improve performance on Pen and Chip. M&M and Card pickup were tasks that did not improve from V1. Both of these tasks require fine manipulation which is still a limitation in V2. Instead, our main improvement from V1 to V2 is in achieving better power grasps.

| ***Hand design*** | *v1* | *v2* | *v3* | *v4* | *v5* |
|---|---|---|---|---|---|
| Palm size | 94x102 | 84x84 | 84x84 | 84x84 | 84x84 |
| Finger length | 90 | 100 | 100 | 100 | 100 |
| MCP diameter | 6 | 6 | 6 | 10 | 8 |
| MCP height | 6 | 8 | 8 | 8 | 8 |
| DIP crease width | 10.3 | 10.3 | 8.9 | 10.3 | 13.0 |
| Thumb angle | 45° | 45° | 0° | 22.5° | 22.5° |
| Fingertip edge | 3.5 | 3.5 | 1.73 | 3.5 | 3.5 |
| Fingertip thickness | 13.21 | 13.22 | 7.98 | 11.22 | 8.75 |
| Finger strength | 37.8 | 47.6 | 34.5 | 51.8 | 27.4 |

Table 5.3: Hand design parameters where finger length refers to the distances in millimeters from the top of the MCP joint to the fingertip and finger strength (N) is measured by pulling on a fully curled finger with a digital force gauge.

Tripod grasps or using more than two fingers was necessary to have stable grasps, especially for the holding tasks such as Hammer, Screwdriver, and Chopstick. However, observations during teleoperation included difficulty using precision grasps with two fingers.

V2 performs better than V1 in 14 tasks (refer to Figure 5.6), including tasks involving Soft ball, Screw driver, Tennis ball, Dry-erase board eraser, and Spam box that all require power grasps. As shown in Figure 5.5, the most significant improvements are seen for Pen, Chip, Tennis ball and Cube Flip. The inset in Figure 5.5 shows V2 grasping Chip with the ring finger and thumb finger, and V2 succeeding at all five repetitions of Cube Flip. These improvements are possible with better reachability and opposability of the thumb with the rest of the fingertips.

Having more space between the fingers made abduction-adduction tasks such as picking Grape off of a stem and Wooden cylinder easier for V1 compared to V2, but V2 still performs reasonably well. Out of the 150 repetitions, V2 is successful in 123 repetitions, which is 18 more when compared to V1. Additionally, the number of tasks where all five repetitions were successful increased from 10 tasks using V1 to 14 tasks using V2.

### 5.4.3 Iteration V3

DASH-V3 has the best thumb opposability and thinnest fingertip design out of all of our hand iterations, yielding in the best score for Card Pickup as shown in Figure 5.5. Thinner fingertips, however, led to weaker finger strength which decreased task success for tasks such as Dry-erase Board Eraser and Grape off stem.

#### Design & Fabrication

The changes from DASH-V2 to V3 involve changing the thumb placement and fingertip shape. In order to make grasps with only two or three fingers more stable, the thumb has to be directly opposable to the rest of the fingers, most importantly the index finger. In V2, the thumb has a 45-degree angle to the palm which we change to be parallel to the index finger in V3, as shown

in the middle of Figure 5.2(b) and in Table 5.3. In addition to the thumb placement, the fingertip shape was changed from a rounded surface to a thinner wedge-like surface (see Figure 5.2). The rounded surface in V2 presented a point contact when interacting with objects. In contrast, the wedge-like surface will have a larger contact area and thinner fingertip (similar to fingernail) in order to get under objects to grasp. This results in a thin fingertip edge, almost half the size of V1 and V2's fingertip edge (see Table 5.3). We also move the tendon routing farther away from the center axis of the MCP joint so that we can exert more torque when folding the finger forward about the MCP joint.

Designing, printing, and assembling V3 took 4, 67.25, and 4.25 hours, respectively. Similar to V2, we reprinted the motor housing again due to the new thumb placement. In total, making V3 took almost 83.75 hours.

#### Evaluation

As shown in Figure 5.6, DASH-V3 has more successful tasks than the previous hand iterations and our baseline, completing 16 tasks successfully in all repetitions as opposed to the 14 tasks V2 successfully executed. V3 succeeded on all repetitions of Wooden Cylinder, Card Pickup, Cup Pouring, Drill, Plush Dinosaur, and Mustard Bottle, which are tasks V2 did not master. The task improvement was due to better thumb opposability compared to V2. In total, V3 succeeded on 124 repetitions which is 1 more than the number of repetitions V2 is successful at. With V3, we observe higher grasp stability during power grasps and handling of delicate objects, during teleoperation. Additionally, we find that the fingertip shape makes a large difference for specific tasks. We clearly see this effect occurring in Cube flip and Card pickup (see inset images of V2 Cube Flip and V3 Card Pickup in Figure 5.5). The flat fingertips of V3 are ideal for thin delicate card pickup but not for the cube flip. Reorienting the cube in-hand in Cube flip is better suited to the rounded fingertip on V2, keeping a stable point contact while the object rotates on the table.

### 5.4.4 Iteration V4

DASH-V4 was optimized for strength as we found that lacking for tasks such as Cube Flip for V3. This allowed for heavy objects like Soft Ball to have great success with V4 as shown in Figure 5.5 but decreased finger folding motion resulted in decreased performance for tasks such as Hammer, Screwdriver, and M&M.

#### Design & Fabrication

The fourth iteration of DASH was designed to optimize for strength. We focused on redesigning the MCP joint to be thicker, providing increased stiffness for folding the fingers into the palm. Achieving the right balance was challenging, as we aimed to maintain the range of motion for MCP forward motion within the torque limits of our motors. While a simple solution would be to use larger motors to increase force and stiffness at the MCP joint, this would result in a larger and heavier hand.

Additionally, we made changes to the fingertips and thumb placement, creating a hybrid design influenced by DASH-V2 and DASH-V3. Thicker fingertips proved useful for tasks in-

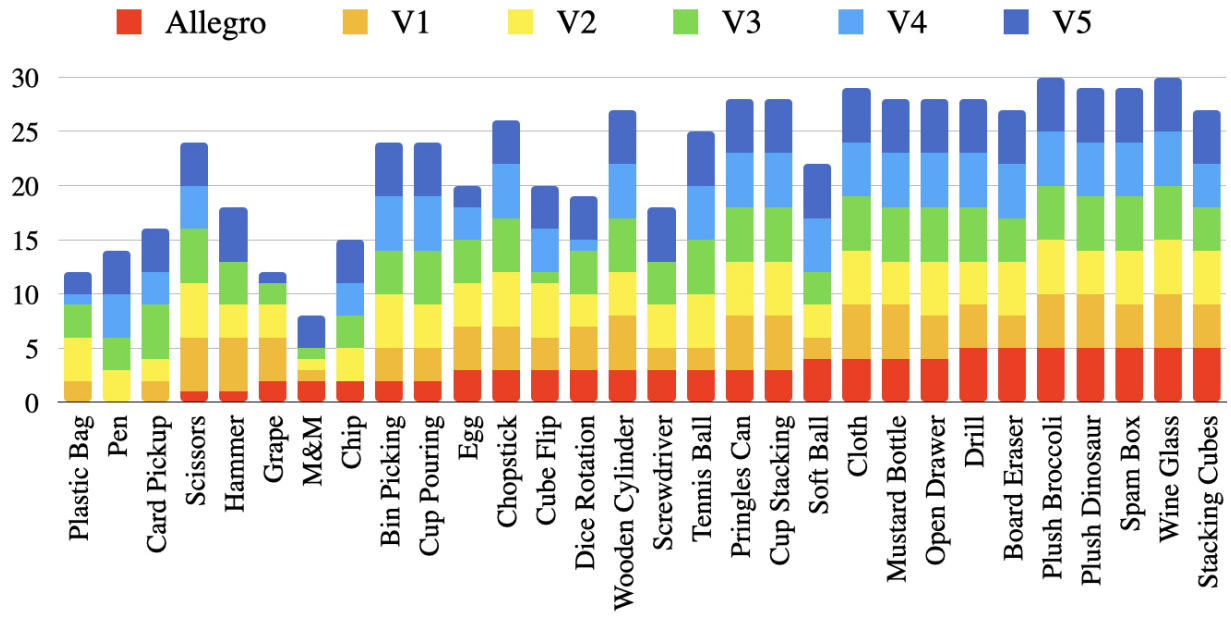



Figure 5.6: Task performance over 5 repetitions of each task across V1, V2, V3, V4, V5, and Allegro as baseline. The tasks are ordered difficult to easy from left to right, according to task performance of Allegro.

volving rotation, such as Cube flipping, while thinner fingertips were beneficial for pinch grasps like Card pickup. The result was rounded edges with a flat surface in the center of the fingertip, providing versatility for pinch to power grasps. Similarly, the thumb placement was positioned between V2 at $45°$ and V3 at $0°$, settling at $22.5°$ relative to the palm. While V2 excelled in power grasps and V3 in pinch grasps like Card pickup, we aimed for V4 to perform equally well in both types of grasping.

Designing, printing, and assembling V4 took 8.5, 82, and 5 hours, respectively. Similar to V3, we reprinted the motor housing to accommodate the new thumb placement.

### Evaluation

DASH-V4 successfully completed all five repetitions of 17 tasks, surpassing the task performance of V3. V4 maintained its performance in most of these tasks, with the exception of Scissors, as shown in Figure 5.5. However, it outperformed V3 in tasks involving the Dry-erase board eraser and performed better than any previous hand iteration in the Soft ball task. This was attributed to the stronger MCP joint, which enhanced the finger strength, as indicated in Table 5.3. Nevertheless, the limited range of motion in the MCP forward joint resulted in poor reachability, causing objects like Scissors to slip between the fingertips.

Overall, the hybrid thumb position and fingertip shape, combining features from V2 and V3, proved advantageous in achieving a greater number of tasks. However, the next iteration should address the loss of range of forward folding motion to improve reachability. The limited reachability of V4 also led to zero successes out of five repetitions in four tasks, including Hammer,

Screwdriver, M&M, and Grape off stem. All of these limitations can be attributed to the restricted range of motion in the MCP forward joint.

### 5.4.5 Iteration V5

DASH-V5 aimed to be a combination of all previous hand design features with respect to joint and fingertip thicknesses. V5 generally outperformed all previous design iterations and excelled at the Screwdriver task as shown in Figure 5.1.

#### Design & Fabrication

The fifth iteration of DASH features a stiffer MCP joint compared to V3, but it is more compliant than the MCP joint of V4. By increasing the compliance at the MCP joint, we were able to achieve a greater range of motion at the joint compared to V4, which had limited folding capabilities. Furthermore, we made the fingertip thinner than that of V4, and widened the DIP crease (as shown in Table 5.3), in order to improve the curling of the finger. As a result, DASH-V5 exhibits the most extensive curling motion among all the previous iterations.

Designing, printing, and assembling V5 took 2, 24, and 2.75 hours, respectively. Unlike the previous versions, we kept the motor assembly unchanged and only replaced the fingers of V4. Consequently, the total time required for iteration was the lowest for V5, totaling 28.75 hours.

#### Evaluation

Among all the design iterations of DASH, DASH-V5 performed the best. V5 succeeded on five out of five repetitions on 19 tasks and achieved a completion rate of 131 out of 150 total task repetitions. In addition to the tasks that V4 succeeded on, V5 also completed five out of five repetitions on the Hammer and Stacking cubes tasks. This improvement indicates that we have made incremental progress on the hand design. V5 had the most curling range of motion than previous hands which made picking objects easier for the teleoperating user due to stable grasps enveloping objects into the palm.

As shown in Figure 5.5, V5 showed improved task performance for Hammer, Screwdriver, Chip, M&M, Grape off stem, and Plastic Bag. However, it performed worse for the Chopsticks and Egg tasks. Although V5 has the lowest finger strength among all DASH iterations due to thinner joints and thinner fingertips (as shown in Table 5.3), its enhanced finger curling abilities even enabled a single finger to hold objects. However, when completely curled, thin objects such as the chopsticks were prone to falling between the thumb and fingers. This issue could be addressed by introducing longer fingers to allow for more overlap between the fingertips. Overall, V5 outperformed all previous iterations of DASH across all 30 tasks. However, it is worth noting that certain hands may specialize in specific tasks. For instance, V5 excelled at picking up Screwdriver, while V3 was the most suitable for Card pickup, as shown in Figure 5.5. One interesting result involved the V5 screwdriver hold, which aligned perfectly in the groove on the tool handle.

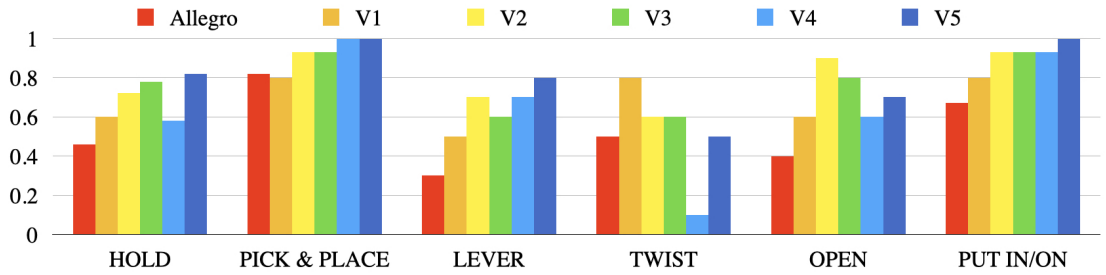



Figure 5.7: Task performance across V1, V2, V3, V4, V5, and Allegro as baseline on each category of tasks from Table 5.2.

## 5.5 Baseline Study: Allegro Dexterous Hand

Allegro [2] is an off-the-shelf gripper that we use as a baseline. Allegro is a dexterous robotic hand that has four fingers with motors at the joints, rigid structure, and large rubber spherical fingertips. We perform the same 30 manipulation tasks from DASH-30 (Table 5.2) with Allegro to compare the performance against all iterations of DASH.

Allegro succeeded on all five repetitions on 7 out of 30 tasks. These tasks included manipulating the Drill, Dry-Erase board eraser, Plush broccoli, Plush Dinosaur, Spam box, Wine glass, and Stacking cubes, as shown as the rightmost tasks in Figure 5.6. Allegro performed best on pick and place tasks compared to other types of tasks as shown in Figure 5.7. However, all iterations of DASH, except V1, were also successful at these tasks.

The Allegro robotic hand and fingers had difficulty with tasks such as picking up the Pen, Card, Plastic bag, Scissors, and Hammer which required precision.While both DASH and Allegro hands lack sensing capabilities, this disproportionately affected Allegro because lack of compliance made it easy to grasp too tightly or not enough, especially for rigid objects. Similarly, the Cup pouring grasp was unstable due to the spherical fingertips rotating the cup in-hand during the task. The side-to-side motion (or abduction-adduction) of the fingers was limited, making Dice Rotation coarse and unpredictable. However, Allegro had stable grasps for larger and softer objects such as the Drill, Softball, Plush Dinosaur, Plush Broccoli, and Wine Glass (see rightmost tasks in Figure 5.6).

## 5.6 Discussion

Across the 30 tasks, we observe that V5 has the best performance solving 19 tasks successfully completing all repetitions, while V4, V3, V2, V1, and Allegro solve 17, 16, 14, 10, and 7 tasks, respectively. In Figure 5.7, we see that all iterations of DASH outperform the Allegro baseline on most categories of tasks listed in Table 5.2 as well as steady improvement in DASH iterations except for twisting and opening which are the most difficult categories of tasks. The two twisting tasks were Dice Rotation and Grape which both more successful with the larger palm and space between fingers for V1 compared to other iterations. For opening tasks, V2 had more success on opening Plastic Bag due to its rounder fingertips and higher finger strength. Through our suite of varied manipulation tasks and human-in-the-loop design iteration, we validate our

framework's ability to use real world evaluation to iteratively design soft robot hands through rapid prototyping and teleoperation.

From our case study in Section 5.4, we draw three crucial observations regarding our proposed framework. Firstly, the direct feedback from the designer performing real world manipulation tasks with DASH was crucial for us in informing the design changes required to improve performance across iterations. In contrast, testing in simulation can result in design changes that do not necessarily translate to performance improvement in the real world. Secondly, using teleoperation removed the necessity of designing different control policies for $30$ various tasks across six robot hand morphologies in our case study, and allowed us to adjust grasps in real-time during task execution, which is often not feasible in simulation or by using keyframed poses. Lastly, despite using real robot hands in the design iteration process, our framework has a short iteration time, consisting mostly of printing time (about $80\%$ of total time), by leveraging 3D-printing and the use of teleoperation to evaluate the design in the real world.

Our framework can extend to testing other soft robotic hands in the real world for rapid design iteration. There are three stages of our framework, as shown in Figure 5.2(a), including design, fabrication, and evaluation. Some best practices include incorporating a modular design to facilitate easier iteration, adopting rapid prototyping methods for seamless fabrication, and favoring incremental design changes to allow for targeted iteration on specific design features. Our method of evaluating using teleoperation also allowed for minimal changes in control when the hand design changed. Our framework can be used to test easily prototyped hands, such as those by Bauer et al. [7] or RBO [35], using the same setup used for DASH iteration, similar to our Manus [98] VR teleoperation system. Additionally, DASH-30, our suite of 30 varied manipulation tasks can be used to benchmark other dexterous hands in the community.

Observing that our robot hand has similar structure and size to human hands, we note a crucial limitation of our framework, shown in Figure 5.2(a), for robot hand morphologies that diverge from human hand morphology as teleoperation might not be feasible in such cases. Additionally, calibration or mapping of the teleoperator's hand to the robot hand can have a significant impact on the robot hand's performance in real-world manipulation tasks. For example, an inaccurate mapping from the teleoperator's hand to the robot hand can incorrectly evaluate the robot hand to be incapable of some tasks. Another limitation for this framework is that it can result in longer turnover times for designs that cannot be made with rapid prototyping techniques such as 3D-printing. Lastly, monotonic improvement is difficult to guarantee due to the manual design iteration process in our framework.

## 5.7 Conclusion and Future Work

This paper presents a design iteration process that can supplement existing design iteration techniques by leveraging 3D-printing and teleoperation. We exhibit the potential of this framework through a case study of designing a 16-DoF 3D-printed dexterous anthropomorphic soft hand DASH. By 3D-printing the new design at each iteration, and evaluating it on real-world manipulation tasks using teleoperation to inform future hand designs, we consistently improve its performance over the baseline Allegro hand and across successive iterations of DASH. We open-sourced our DASH CAD models and teleoperated demonstration data at

https://dash-through-interaction.github.io.

Future directions include automatic design iteration by singling out features of the CAD design and correlating them with capabilities of the hand. Further study would be required to automate this process and use collected data to learn what properties of the hand should be improved for better task performance. Currently, the process of design iteration in our case study was manual in that we chose parameters to change based on task performance and observations from real-world manipulation experiments.

# Chapter 6

# Evaluating Soft Hands using Learned Policies

**Paper:** *Kannan, A., Shaw, K., Bahl, S., Mannam, P., & Pathak D. (2023). DEFT: dexterous fine-tuning for hand policies. In 7th Annual Conference on Robot Learning (CORL).*

## 6.1 Introduction

The longstanding goal of robot learning is to build robust agents that can perform long-horizon tasks autonomously. This could for example mean a self-improving robot that can build furniture or an agent that can cook for us. A key aspect of most tasks that humans would like to perform is that they require complex motions that are often only achievable by hands, such as hammering a nail or using a screwdriver. Therefore, we investigate dexterous manipulation and its challenges in the real world.

A key challenge in deploying policies in the real world, especially with robotic hands, is that there exist many failure modes. Controlling a dexterous hand is much harder than end-effectors due to larger action spaces and complex dynamics. To address this, one option is to *improve* directly in the real world via *practice*. Traditionally, reinforcement learning (RL) and imitation learning (IL) techniques have been used to deploy hands-on tasks such as in-hand rotation or grasping. This is the case as setups are often built so that it is either easy to simulate in the real world or robust to practice. However, the real world contains tasks that one cannot simulate (such as manipulation of soft objects like food) or difficult settings in which the robot cannot practice (sparse long-horizon tasks like assembly). How can we build an approach that can scale to such tasks?

There are several issues with current approaches for practice and improvement in the real world. Robot hardware often breaks, especially with the amount of contact to learn dexterous tasks like operating tools. We thus investigate using a *soft anthropomorphic hand* [101], which can easily run in the real world without failures or breaking. This soft anthropomorphic hand is well-suited to our approach as it is flexible and can gently handle object interactions. The hand does not get damaged by the environment and is robust to continuous data collection. Due to its

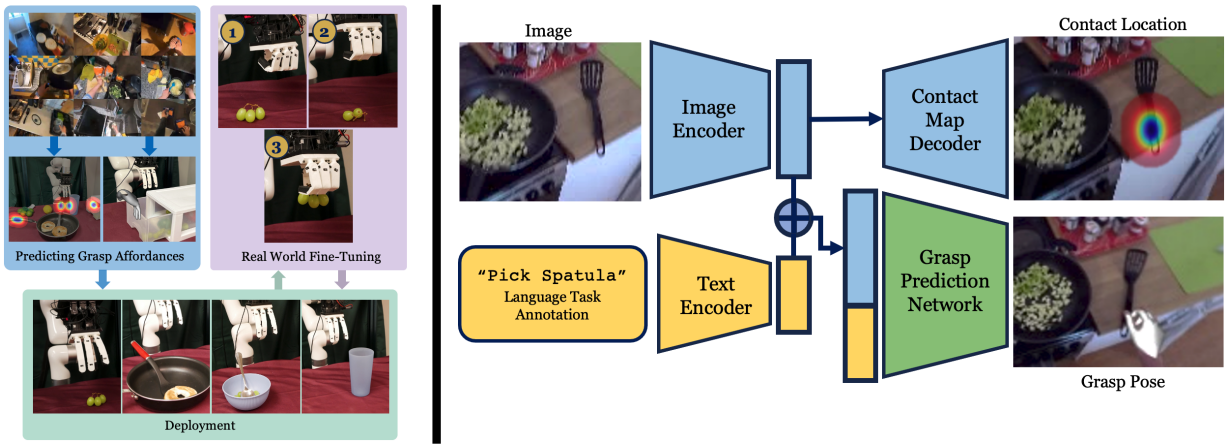



Figure 6.1: **Left:** DEFT consists of two phases: an affordance model that predicts grasp parameters followed by online fine-tuning with CEM. **Right:** Our affordance prediction setup predicts grasp location and pose.

human-like proportions and morphology, retargeting human hand grasps to robot hand grasps is made simpler.

Unfortunately, this hand is difficult to simulate due to its softness. Directly learning from scratch is also difficult as we would like to build *generalizable policies*, and not practice for every new setting. To achieve efficient real-world learning, we must learn a prior for reasonable behavior to explore using useful actions. Due to recent advances in computer vision, we propose *leveraging human data to learn priors* for dexterous tasks, and improving on such priors in the real world. We aim to use the vast corpus of internet data to define this prior. What is the best way to combine human priors with online practice, especially for hand-based tasks? When manipulating an object, the first thing one thinks about is where on the object to make contact, and how to make this contact. Then, we think about how to move our hands *after the contact*. In fact, this type of prior has been studied in computer vision and robotics literature as *visual affordances* [102, 103, 104, 105, 106, 107, 108, 109]. Our approach, DEFT, builds grasp affordances that predict the contact point, hand pose at contact, and post contact trajectory. To improve upon these, we introduce a sampling-based approach similar to the Cross-Entropy Method (CEM) to fine-tune the grasp parameters in the real world for a variety of tasks. By learning a residual policy [110, 111], CEM enables iterative real-world improvement in less than an hour.

In summary, our approach (DEFT) executes real-world learning on a soft robot hand with only a few trials in the real world. To facilitate this efficiently, we train priors on human motion from internet videos. We introduce 9 challenging tasks (as seen in Figure 7.1) that are difficult even for trained operators to perform. While our method begins to show good success on these tasks with real-world fine-tuning, more investigation is required to complete these tasks more effectively.

## 6.2 Related Work

**Real-world robot learning** Real-world manipulation tasks can involve a blend of classical and learning-based methods. Classical approaches like control methods or path planning often use hand-crafted features or objectives and can often lack flexibility in unstructured settings [112, 113, 114]. On the other hand, data-driven approaches such as deep reinforcement learning (RL) can facilitate complex behaviors in various settings, but these methods frequently rely on lots of data, privileged reward information and struggle with sample efficiency [115, 116, 117, 118, 119]. Efforts have been made to scale end-to-end RL [120, 121, 122, 123, 124, 125] to the real world, but their approaches are not yet efficient enough for more complex tasks and action spaces and are reduced to mostly simple tasks even after a lot of real-world learning. Many approaches try to improve this efficiency such as by using different action spaces [126], goal relabeling [127], trajectory guidance [128], visual imagined goals [121], or curiosity-driven exploration [129]. Our work focuses on learning a prior from human videos in order to learn efficiently in the real world.

**Learning from Human Motion** The field of computer vision has seen much recent success in human and object interaction with deep neural networks. The human hand is often parametrized with MANO, a 45-dimensional vector [130] of axes aligned with the wrist and a 10-dimensional shape vector. MANOtorch from [131] aligns it with the anatomical joints. Many recent works detect MANO in monocular video [86, 132, 133]. Some also detect objects as well as the hand together [106, 134]. We use FrankMocap to detect the hand for this work. There are many recent datasets including the CMU Mocap Database [135] and Human3.6M [136] for human pose estimation, 100 Days of Hands [106] for hand-object interactions, FreiHand [137] for hand poses, Something-Something [138] for semantic interactions. ActivityNet datasets [139], or YouCook [140] are action-driven datasets that focus on dexterous manipulation. We use these three datasets: [141] is a large-scale dataset with human-object interactions, [142] for curated human-object interactions, and [143] which has many household kitchen tasks. In addition to learning exact human motion, many others focus on learning priors from human motion. [144, 145] learn general priors using contrastive learning on human datasets.

**Learning for Dexterous Manipulation** With recent data-driven machine learning methods, roboticists are now beginning to learn dexterous policies from human data as well. Using the motion of a human can be directly used to control robots [87, 100, 146]. Moving further, human motion in internet datasets can be retargeted and used directly to pre-train robotic policies [147, 148]. Additionally, using human motion as a prior for RL can help with learning skills that are human-like [149, 150, 151]. Without using human data as priors, object reorientation using RL has been recently successful in a variety of settings [67, 152]. Similar to work in robot dogs which do not have an easy human analog to learn from, these methods rely on significant training data from simulation with zero-shot transfer [153, 154].

**Soft Object Manipulation** Manipulating soft and delicate objects in a robot's environment has been a long-standing problem. Using the torque output on motors, either by measuring

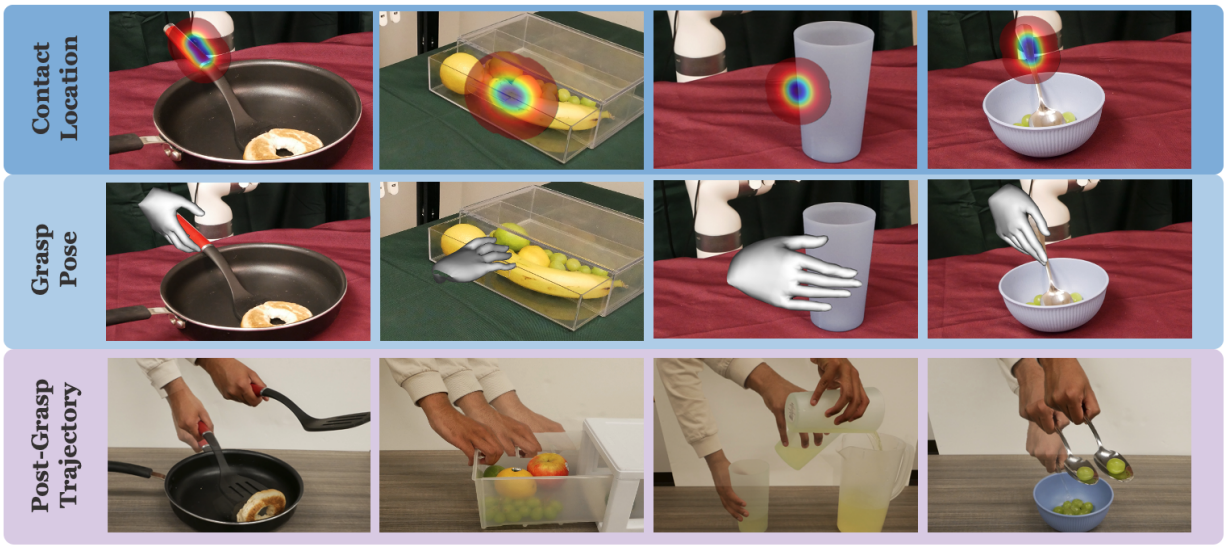



Figure 6.2: We produce three priors from human videos: the contact location (**top row**) and grasp pose (**middle row**) from the affordance prior; the post-grasp trajectory (**bottom row**) from a human demonstration of the task.

current or through torque sensors, is useful feedback to find out how much force a robot is applying [155, 156]. Coupled with dynamics controllers, these robots can learn not to apply too much torque to the environment around them [157, 158, 159]. A variety of touch sensors [160, 161, 162, 163] have also been developed to feel the environment around it and can be used as control feedback. Our work does not rely on touch sensors. Instead, we practice in the real world to learn stable and precise grasps.

## 6.3 Fine-Tuning Affordance for Dexterity

The goal of DEFT is to learn useful, dexterous manipulation in the real world that can generalize to many objects and scenarios. DEFT learns in the real world and fine-tunes robot hand-to-object interaction in the real world using only a few samples. However, without any priors on useful behavior, the robot will explore inefficiently. Especially with a high-dimensional robotic hand, we need a strong prior to effectively explore the real world. We thus train an affordance model on human videos that leverages human behavior to learn reasonable behaviors the robot should perform.

### 6.3.1 Learning grasping affordances

To learn from dexterous interaction in a sample efficient way, we use human hand motion as a prior for robot hand motion. We aim to answer the following: (1) What useful, actionable information can we extract from the human videos? (2) How can human motion be translated to the robot embodiment to guide the robot? In internet videos, humans frequently interact with a wide variety of objects. This data is especially useful in learning object affordances. Furthermore, one of the major obstacles in manipulating objects with few samples is accurately

grasping the object. A model that can perform a strong grasp must learn *where* and *how* to grasp. Additionally, the task objective is important in determining object affordances–humans often grasp objects in different ways depending on their goal. Therefore, we extract three items from human videos: the grasp location, human grasp pose, and task.

Given a video clip $V = \{v_1, v_2, \ldots, v_T\}$, the first frame $v_t$ where the hand touches the object is found using an off-the-shelf hand-object detection model [106]. Similar to previous approaches [103, 104, 105, 107], a set of contact points are extracted to fit a Gaussian Mixture Model (GMM) with centers $\mu = \{\mu_1, \mu_2, \ldots, \mu_k\}$. Detic [164] is used to obtain a cropped image $v_1'$ containing just the object in the initial frame $v_1$ to condition the model. We use Frankmocap [86] to extract the hand grasp pose $P$ in the contact frame $v_t$ as MANO parameters. We also obtain the wrist orientation $\theta_{\text{wrist}}$ in the camera frame. This guides our prior to output wrist rotations and hand joint angles that produce a stable grasp. Finally, we acquire a text description $T$ describing the action occurring in $V$.

We extract affordances from three large-scale, egocentric datasets: Ego4D [141] for its large scale and the variety of different scenarios depicted, HOI4D [108] for high-quality human-object interactions, and EPIC Kitchens [143] for its focus on kitchen tasks similar to our robot's. We learn a task-conditioned affordance model $f$ that produces $(\hat{\mu}, \hat{\theta}_{\text{wrist}}, \hat{P}) = f(v_1', T)$. We predict $\hat{\mu}$ in similar fashion to [103]. First, we use a pre-trained visual model [50] to encode $v_1'$ into a latent vector $z_v$. Then we pass $z_v$ through a set of deconvolutional layers to get a heatmap and use a spatial softmax to estimate $\hat{\mu}$.

To determine $\hat{\theta}_{\text{wrist}}$ and $\hat{P}$, we use $z_v$ and an embedding of the text description $z_T = g(T)$, where $g$ is the CLIP text encoder [165]. Because transformers have seen success in encoding various multiple modes of input, we use a transformer encoder $\mathcal{T}$ to predict $\hat{\theta}_{\text{wrist}}, \hat{P} = \mathcal{T}(z_v, z_T)$. Overall, we train our model to optimize

| Parameter | Dimensions | Description |
|---|---|---|
| $\mu$ | 3 | XYZ grasp location in workspace |
| $\theta_{\text{wrist}}$ | 3 | Wrist grasp rotation (euler angles) |
| $P$ | 16 | Finger joint angles in soft hand |

Table 6.1: Parameters that are fine-tuned in the real world. The affordance model predicts a 45-dimensional hand joint pose for $P$, which is retargeted to a 16-dimensional soft hand pose.

$$\mathcal{L} = \lambda_\mu ||\mu - \hat{\mu}||_2 + \lambda_\theta ||\theta_{\text{wrist}} - \hat{\theta}_{\text{wrist}}||_2 + \lambda_P ||P - \hat{P}||_2 \tag{6.1}$$

At test time, we generate a crop of the object using Segment-Anything [166] and give our model a task description. The model generates contact points on the object, and we take the average as our contact point. Using a depth camera, we can determine the 3D contact point to navigate to. While the model outputs MANO parameters [130] that are designed to describe human hand joints, we retarget these values to produce similar grasping poses on our robot hand in a similar manner to previous approaches [81, 87]. For more details, we refer readers to the appendix.

In addition to these grasp priors, we need a task-specific post-contact trajectory to successfully execute a task. Because it is challenging to learn complex and high-frequency action information from purely offline videos, we collect one demo of the human doing the robot task


Figure 6.3: **Left**: Workspace Setup. We place an Intel RealSense camera above the robot to maintain an egocentric viewpoint, consistent with the affordance model's training data. **Right**: Thirteen objects used in our experiments.

(Figure 6.2) separate from the affordance model $f$. We extract the task-specific wrist trajectory after the grasp using [86]. We compute the change in wrist pose between adjacent timesteps for the first 40 timesteps. When deployed for fine-tuning, we execute these displacements for the post-grasp trajectory. Once we have this prior, how can the robot *improve* upon it?

### 6.3.2 Fine-tuning via Interaction

**Algorithm 1** Fine-Tuning Procedure for DEFT

**Require:** Task-conditioned affordance model $f$, task description $T$, post-grasp trajectory $\tau$, parameter distribution $\mathcal{D}$, residual cVAE policy $\pi$. $E$ number of elites, $M$ number of warm-up episodes, $N$ total iterations.

$\mathcal{D} \leftarrow \mathcal{N}(\mathbf{0}, \sigma^2)$
**for** $k = 1 \ldots N$ **do**
  $I_{k,0} \leftarrow$ initial image
  $\xi_k \leftarrow f(I_{k,0}, T)$
  Sample $\epsilon_k \sim D$
  Execute grasp from $\xi_k + \epsilon_k$, then trajectory $\tau$
  Collect reward $R_k$; reset environment
  **if** $k > M$ **then**
    Order traj indices $i_1, i_2, \ldots, i_k$ based on rewards
    $\Omega \leftarrow \{\epsilon_{i_1}, \epsilon_{i_2}, \ldots, \epsilon_{i_E}\}$
    Fit $\mathcal{D}$ to distribution of residuals in $\Omega$
Fit $\pi(.)$ as a VAE to $\Omega$

The affordance prior allows the robot to narrow down its learning behavior to a small subset of all possible behaviors. However, these affordances are not perfect and the robot will oftentimes still not complete the task. This is partially due to morphology differences between the human and robot hands, inaccurate detections of the human hands, or differences in the task setup. To improve upon the prior, we practice learning a residual policy for the grasp parameters in Table 6.1.

Residual policies have been used previously to efficiently explore in the real world [111, 167]. They use the prior as a starting point and explore nearby. Let the grasp location, wrist rotation, grasp pose, and trajectory from our affordance prior be $\xi$. During training we

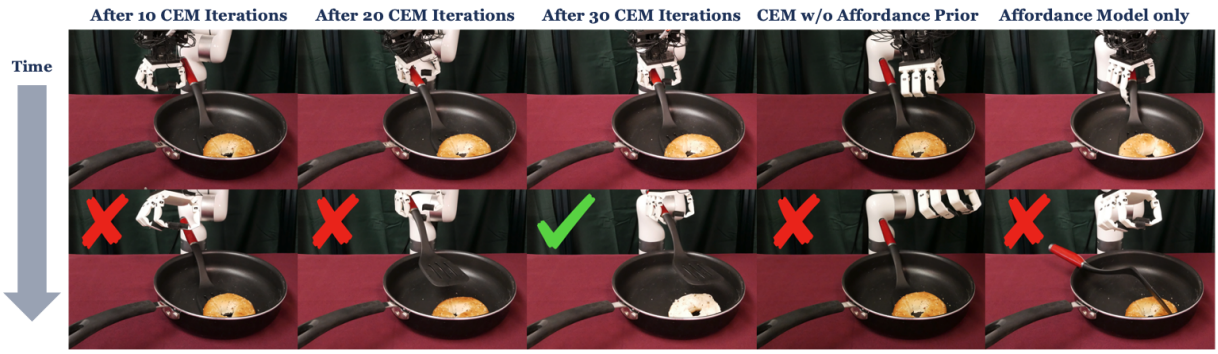



Figure 6.4: Qualitative results showing the finetuning procedure for DEFT. The model learns to hold the spatula and flip the bagel after 30 CEM iterations.

sample noise $\epsilon \sim \mathcal{D}$ where $\mathcal{D}$ is initialized to $\mathcal{N}(0, \sigma^2)$ (for a small $\sigma$). We rollout a trajectory parameterized by $\xi + \epsilon$. We collect $R_i$, the reward for each $\xi_i = f(v_i) + \epsilon_i$ where $v_i$ is the image. First, we execute an initial number of $M$ warmup episodes with actions sampled from $\mathcal{D}$, recording a reward $R_i$ based on how well the trajectory completes the task. For each episode afterward, we rank the prior episodes based on the reward $R_i$ and extract the sampled noise from the episodes with the highest reward (the 'elites' $\Omega$). We fit $\mathcal{D}$ to the elite episodes to improve the sampled noise. Then we sample actions from $\mathcal{D}$, execute the episode, and record the reward. By repeating this process we can gradually narrow the distribution around the desired values. In practice, we use $M = 10$ warmup episodes and a total of $N = 30$ episodes total for each task. This procedure is shown in Algorithm 1.

At test time, we could take the mean values of the top $N$ trajectories for the rollout policy. However, this does not account for the appearance of different objects, previously unseen object configurations, or other properties in the environment. To generalize to different initializations, we train a VAE [168, 169, 170, 171] to output residuals $\delta_j$ conditioned on an encoding of the initial image $\phi(I_{j,0})$ and affordance model outputs $\xi_j$ from the top ten trajectories. We train an encoder $q(z|\delta_j, c_j)$ where $c_j = (\phi(I_{j,0}), \xi_j)$, as well as a decoder $p(\delta_j|z, c_j)$ that learns to reconstruct residuals $\delta_j$. At test time, our residual policy $\pi(I_0, \xi)$ samples the latent $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and predicts $\hat{\delta} = p(z, (I_0, \xi))$. Then we rollout the trajectory determined by the parameters $\xi + \hat{\delta}$. Because the VAE is conditioned on the initial image, we generalize to different locations and configurations of the object.

## 6.4 Experiment Setup

We perform a variety of experiments to answer the following: 1) How well can DEFT learn and improve in the real world? 2) How good is our affordance model? 3) How can the experience collected by DEFT be distilled into a policy? 4) How can DEFT be used for complex, soft object manipulation? Please see our website at `http://dexterous-finetuning.github.io` for videos.

**Task Setup** We introduce 9 tabletop tasks, *Pick Cup*, *Pour Cup*, *Open Drawer*, *Pick Spoon*, *Scoop Grape*, *Stir Spoon*, *Pick Grape*, *Flip Bagel*, *Squeeze Lemon*. Robotic hands are especially well-suited for these tasks because most of them require holding curved objects or manipulat-

| Method | Pick cup | | Pour cup | | Open drawer | | Pick spoon | | Scoop Grape | | Stir Spoon | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | train | test | train | test | train | test | train | test | train | test | train | test |
| `Real-World Only` | 0.0 | 0.1 | 0.2 | 0.1 | 0.1 | 0.0 | 0.7 | 0.3 | 0.0 | 0.0 | 0.3 | 0.0 |
| `Affordance Model Only` | 0.1 | | 0.4 | | **0.5** | | 0.5 | | 0.0 | | 0.3 | |
| **`DEFT`** | **0.8** | **0.8** | **0.8** | **0.9** | **0.5** | **0.4** | **0.8** | **0.6** | **0.7** | **0.3** | **0.8** | **0.5** |

Table 6.2: We present the results of our method as well as compare them to other baselines: Real-world learning without internet priors used as guidance and the affordance model outputs without real-world learning.We evaluate the success of the methods on the tasks over 10 trials.

ing objects with tools to succeed. For all tasks, we randomize the position of the object on the table, as well as use train and test objects with different shapes and appearances to test for generalization. To achieve real-world learning with the soft robot hand, we pretrain an internet affordance model as a prior for robot behavior. As explained in Section 6.3, we train one language-conditioned model on all data. At test time, we use this as initialization for our real-world fine-tuning. The fine-tuning is done purely in the real world. An operator runs 10 warmup episodes of CEM, followed by 20 episodes that continually update the noise distribution, improving the policy. After this stage, we train a residual VAE policy that trains on the top ten CEM episodes to predict the noise given the image and affordance outputs. We evaluate how effectively the VAE predicts the residuals on each of the tasks by averaging over 10 trials. Because it takes less than an hour to fine-tune for one task, we are able to thoroughly evaluate our method on 9 tasks, involving over 100 hours of real-world data collection.

**Hardware Setup** We use a 6-DOF UFactory xArm6 robot arm for all our experiments. We attach it to a 16-DOF Soft Hand using a custom, 3D-printed base. We use a single, egocentric RGBD camera to capture the 3D location of the object in the camera frame. We calibrate the camera so that the predictions of the affordance model can be converted to and executed in the robot frame. The flexibility of the robot hand also makes it robust to collisions with objects or unexpected contact with the environment. For the arm, we ensure that it stays above the tabletop. The job will be terminated if the arm's dynamics controller senses that the arm collided aggressively with the environment.

## 6.5 Results

**Effect of affordance model** We investigate the role of the affordance model and real-world fine-tuning (Table 6.2 and Figure 6.5). In the real-world only model, we provide a few heuristics in place of the affordance prior. We detect the object in the scene using a popular object detection model [166] and let the contact location prior be the center of the bounding box. We randomly sample the rotation angle and use a half-closed hand as the grasp pose prior. With these manually provided priors, the robot has difficulty finding stable grasps. The main challenge was finding the correct rotation angle for the hand. Hand rotation is very important for many tool manipulation tasks because it requires not only picking the tool but also grasping in a stable manner.

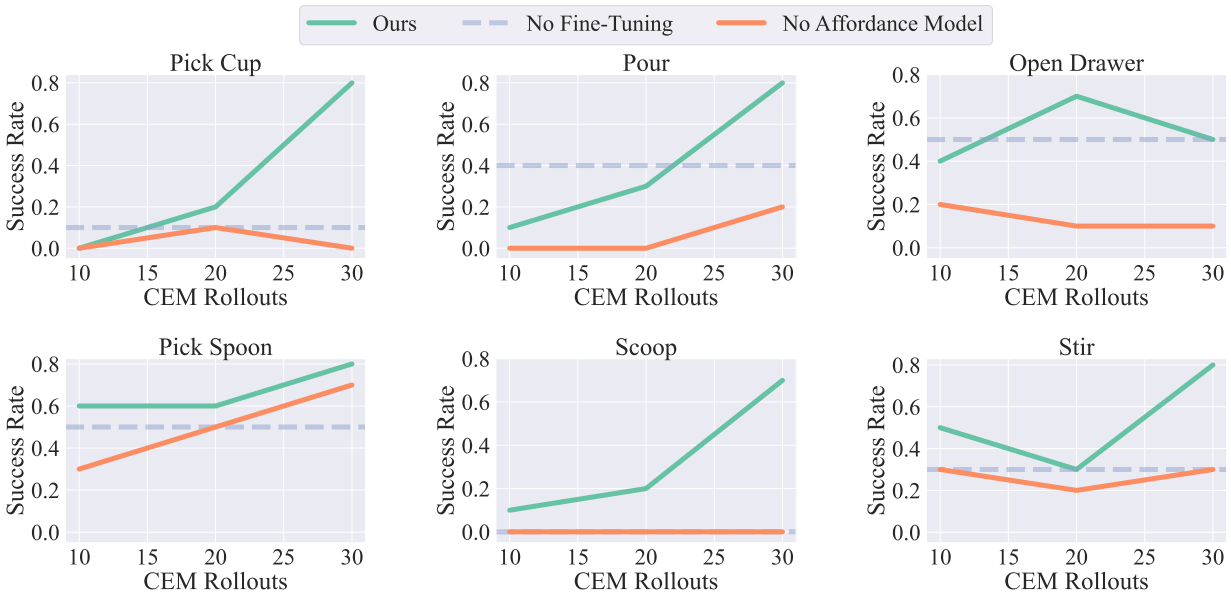



Figure 6.5: Improvement results for 6 tasks: pick cup, pour, open drawer, pick spoon, scoop, and stir. We see a steady improvement in our method as more CEM episodes are collected.

**Zero-shot model execution** We explore the zero-shot performance of our prior. Without applying any online fine-tuning to our affordance model, we rollout the trajectory parameterized by the prior. While our model is decent on simpler tasks, the model struggles on tasks like stir and scoop that require strong power grasps (shown in Table 6.2). In these tasks, the spoon collides with other objects, so fine-tuning the prior to hold the back of the spoon is important in maintaining a reliable grip throughout the post-grasp motion. Because DEFT incorporates real-world experience with the prior, it is able to sample contact locations and grasp rotations that can better execute the task.

**Human and automated rewards** We ablate the reward function used to evaluate episodes. Our method queries the operator during the task reset process to assign a continuous score from 0 to 1 for the grasp. Because the reset process requires a human-in-the-loop regardless, this adds little marginal cost for the operator. But what if we would like these rewards to be calculated autonomously? We use the final image collected in the single post-grasp human demonstration from Section 6.3 as the goal image. We define the reward to be the negative embedding distance between the final image of the episode and the goal image with either an R3M [50] or a ResNet [172] encoder. The model learned from ranking trajectories with R3M reward is competitive with DEFT in all but one task, indicating that using a visual reward model can provide reasonable results compared to human rewards.

**Model Architecture** We investigate different models and training architectures for the policy trained on the rollouts (Table 6.3).

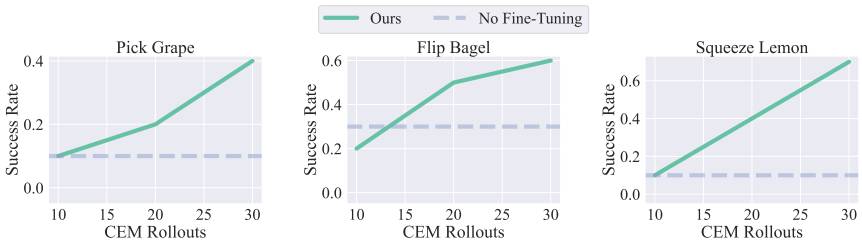



Figure 6.6: We evaluate DEFT on three difficult manipulation tasks.

| Method | Pour Cup | | Open Drawer | | Pick Spoon | |
|---|---|---|---|---|---|---|
| | train | test | train | test | train | test |
| *Reward Function:* | | | | | | |
| **R3M Reward** | 0.0 | 0.0 | 0.4 | **0.5** | 0.5 | 0.4 |
| **Resnet18 Imagenet Reward** | 0.1 | 0.2 | 0.3 | 0.1 | 0.4 | 0.2 |
| *Policy Ablation:* | | | | | | |
| **DEFT w/ MLP** | 0.0 | 0.0 | 0.5 | 0.0 | 0.6 | 0.5 |
| **DEFT w/ Transformer** | 0.4 | 0.5 | **0.6** | 0.1 | 0.4 | 0.5 |
| **DEFT w/ Direct Parameter est.** | 0.1 | 0.1 | 0.1 | 0.0 | 0.3 | 0.0 |
| **DEFT** | **0.8** | **0.9** | 0.5 | 0.4 | **0.8** | **0.6** |

Table 6.3: Ablations for (1) reward function type, (2) model architecture, and (3) parameter estimation.

When we replace the conditional VAE with an MLP that predicts residuals, the model has difficulty learning the grasp rotation to effectively pour a cup. We find that the MLP cannot learn the multimodality of the successful data properly. Our transformer ablation is an offline method similar to [173] where in addition to the image and affordance model outputs, we condition on the reward outputs and train a transformer to predict the residual. At test time the maximum reward is queried and the output is used in the rollout. While this method performs well, we hypothesize that the transformer needs more data to match DEFT. Finally, we train a VAE to directly estimate $\xi$ instead of the residual. This does not effectively distill the information from the affordance prior without the training time allotted. As a result, it often makes predictions that are far from the correct grasp pose.

**Performance on complex tasks and soft object manipulation** We investigate the performance of DEFT on more challenging tasks. Tasks involving soft objects cannot be simulated accurately, while our method is able to perform reasonably on food manipulation tasks as shown in Figure 6.6.

Of the three tasks, our method has the most difficulty with the Pick Grape task. Because grapes are small, the fingers must curl fully to maintain a stable grasp. A limitation of our hand is that the range of its joints does not allow it to close the grasp fully and as a result, it has difficulty in consistently picking small objects. This also makes it challenging to hold heavy objects like the spatula in Flip Bagel, but with practice DEFT learns to maintain a stable grasp of the spatula. For Squeeze Lemon, DEFT develops a grasp that allows it to apply sufficient pressure above the juicer. Specifically, our method takes advantage of the additional fingers available for support in hands.

## 6.6 Discussion and Limitations

In this paper, we investigate how to learn dexterous manipulation in complex setups. DEFT aims to learn directly in the real world. In order to accelerate real-world fine-tuning, we build an *affordance* prior learned from human videos. We are able to efficiently practice and improve in

the real world via our online fine-tuning approach with a soft anthropomorphic hand, performing a variety of tasks (involving both rigid and soft objects). While our method shows some success on these tasks, there are some limitations to DEFT that hinder its efficacy. Although we are able to learn policies for the high-dimensional robot hand, the grasps learned are not very multi-modal and do not capture all of the different grasps humans are able to perform. This is mainly due to noisy hand detections in affordance pretraining. As detection models improve, we hope to be able to learn a more diverse set of hand grasps. Second, during finetuning, resets require human input and intervention. This limits the real-world learning we can do, as the human has to be constantly in the loop to reset the objects. Lastly, the hand's fingers cannot curl fully. This physical limitation makes it difficult to hold thin objects tightly. Future iterations of the soft hand can be designed to grip such objects strongly.

# Chapter 7

# Automating Design of Soft Hands

**Paper (under review):** *Mannam, P., Liu, X., Zhao D., Oh, J., & Pollard, N. (2024). Automating design iteration for dexterous anthropomorphic soft robotic hands. In 7th IEEE-RAS International Conference on Soft Robotics (RoboSoft).*

## 7.1 Introduction

Humans have evolved to master manipulation of human-designed objects and tools. Similarly, robot hands have been designed over many years to progress towards human-like dexterity and manipulation skills [11]. While multi-fingered robotic hands can vary in degrees-of-freedom, actuation, and more, evaluation criteria of hands has been limited to grasp taxonomies [19] and manipulation task success [27, 29, 174]. While some suite of tasks aim to evaluate dexterous manipulation [28, 175, 176], it is difficult to learn dexterous skills for new hand designs which limits the utilization of dexterous skill benchmarks for evaluation. Thus, we want to efficiently learn dexterous manipulation skills on new hand morphologies in order to evaluate numerous designs rapidly for design iteration.

Testing robot hands on their downstream tasks, such as picking up a cellphone from the table, can be difficult to execute for hundreds of designs since each design requires a different control policy. However, recent works have shown that transferring control policies across evolutions of robotic designs from a source to target robot improves efficiency for learning new policies [177, 178]. Continuous evolutionary models allow for transferring policies from expert policies on source robots to intermediate robots instead of learning new policies for target robots. While these approaches focus on policy optimization, we show that we can apply this framework to design and policy co-optimization for automating design iteration of robotic hands.

Using genetic algorithms and robot interpolation, we can explore dexterous hand designs for a set of manipulation tasks. Genetic algorithms have been used for optimization problems including mobile robots and trajectory planning [179, 180, 181]. Robotic interpolation allows for finding the morphology and kinematics of intermediate robots [177, 178]. We utilize both of these concepts to generate new robotic hand designs and learn control policies for them in order to find an optimized top-performing robotic hand design.

Through learning dexterous manipulation skills on hundreds of robotic hand designs in sim-

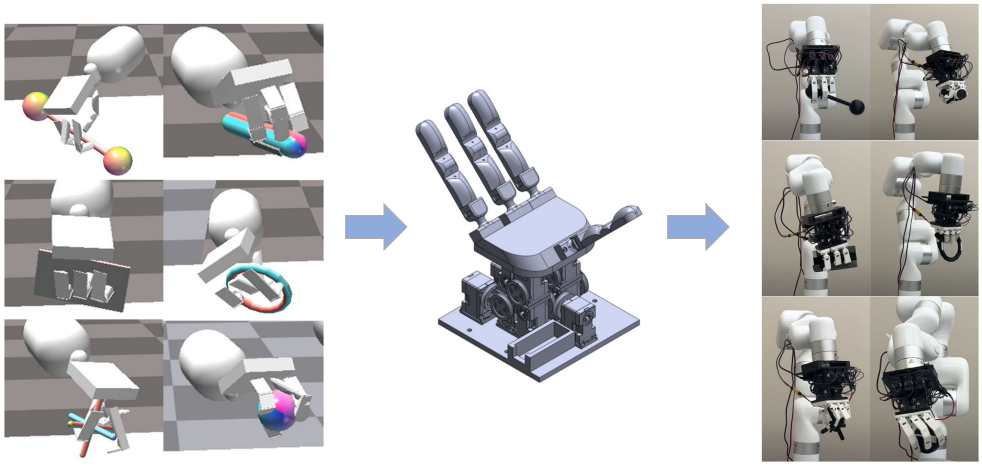



Figure 7.1: Automated design iteration for dexterous soft hands: a) design optimization of robotic hand designs in simulation using genetic algorithms and policy transfer, b) fabrication of optimized hand that outperforms other hand designs in simulation, and c) design evaluation of optimized hand in the real-world using teleoperation on the same set of manipulation tasks.

ulation, we can automate the design iteration of our robotic hands. As shown in Figure 7.1a, we use genetic algorithms to start from a set of hand design candidates from prior work [101], allowing for crossovers and mutations to exploit and explore design features, such as finger arrangement and finger length, that allow for successful manipulation of six different objects in pick up and reorient manipulation tasks. These dexterous manipulation tasks allow for differentiating hand design performance among 396 anthropomorphic hand designs that were generated. Subsequently, we fabricate the best-performing hand designs from simulation using 3D-printing (Figure 7.1b). Using the same six objects in the real world, we use teleoperation to pick up and reorient the objects in-hand, as depicted in Figure 7.1c, to evaluate whether the optimized hand designs can succeed at the dexterous manipulation tasks.

Our hypothesis that the trend in performance of these hand designs in simulation resembles real world performance is supported by our results from more than 900 real-world teleoperated manipulation experiments. In addition, our design and policy co-optimization approach results in two optimized soft robot hand designs, from simulation, that outperform existing soft hand designs, from prior work [101], in real world evaluation, despite the considerable sim-to-real gap. In summary, the contributions of this work include:

- An approach for design and policy co-optimization for soft robot hands using genetic algorithm and policy transfer in simulation.
- Generation of new hand designs, using the above approach, in simulation that can be fabricated in the real world as tendon-driven soft hands.
- Teleoperated evaluation of two optimized fabricated hand designs in the real-world to show that they outperform existing soft hands from prior work [101].

## 7.2 Related Work

Design and control of a robot are intrinsically linked. Hence, jointly co-optimizing design and control policies of robots can be done to avoid determining control policies for every single design candidate. This idea has been explored in a wide range of areas extending from locomotion [32] to manipulation [182]. Recent work have leveraged data-driven and learning-based approaches for co-optimization using concurrent networks [183]. Co-optimization can also be posed as an optimal control problem and has been applied to both manipulation and locomotion using this formulation [184]. Similarly, reinforcement learning can be used to evolve legged robots and their gait towards an optimal design and control policy [185, 186]. Our approach also uses a reinforcement learning based approach to co-optimize both the design of an anthropomorphic soft robot hand and a control policy for dexterous manipulation tasks.

For design and control co-optimization of robot hands, previous works have used gradient-based approaches and evolutionary methods [182]. Xu et. al. [187] show that using gradient-based optimization methods outperform both gradient-free methods and model-free reinforcement learning approaches for co-optimization. On the other hand, evolutionary methods are a scalable solution to reinforcement learning [188]. Continuous robot evolution and human demonstrations can transfer control policies from a five-fingered dexterous robot hand to a two-finger gripper [178]. Furthermore, evolutionary algorithms can be used to co-optimize hand design and control for any manipulation task through optimization of joint limit parameters for increased robustness [44]. While this approach aims to find the simplest hand design for the task, targeting grasping strategies for specific manipulators,like a soft hand, can adapt existing control strategies or invent new control strategies to co-optimize design and control [43]. In a similar fashion, our approach uses genetic algorithm for exploring new hand designs and policy transfer to learn new policies for hand designs efficiently in simulation.

Optimizing on robotic hand designs in simulation can pose difficulties for sim-to-real transfer when executing the control policies in the real world. Instead, we use teleoperation setups similar to those previously used for imitation learning data collection [92, 93]. Our real world evaluation setup is similar to our previous work [101], where we use teleoperation to evaluate the capabilities and limitations of our soft robot hands and their designs. However, our prior work used manual design iteration and did not leverage simulation to optimize hand designs. Our manipulation task setup is similar to the tool positioning task used for chaining multiple dexterous tasks for long-horizon task goals [189], where an object is first grasped from the table and then reoriented in-hand to a final desired pose. Contrary to [189], we evaluate the same task in the real-world using teleoperation for our generated hand designs. Finally, our approach uses a qualitiative metric, similar to one used in [190], that captures the quality of the grasp for evaluating the teleoperated manipulation tasks for each of the optimized hand designs in the real world.

## 7.3 Dexterous Anthropomorphic Soft Hand

Our robotic hand designs are simulated as rigid robotic hands but fabricated as soft tendon-driven hands using 3D-printing. We implement the robotic hand designs as soft robots for safety

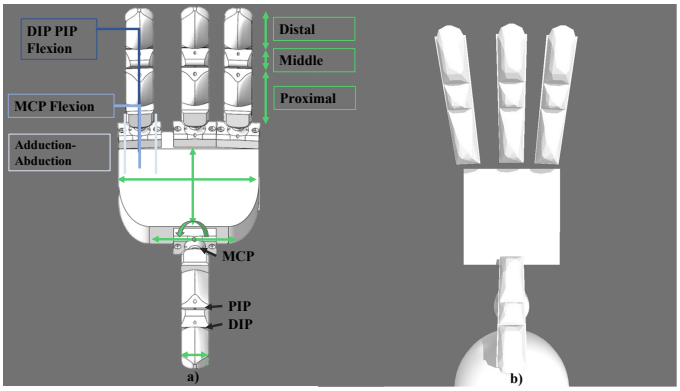



Figure 7.2: a) CAD of top-performing optimized hand V7 shown with DIP, PIP, and MCP joints labelled, as well as hand design parameters shown with green arrows. Tendon placements along finger are shown in blue where each controls MCP adduction-abduction movement, MCP forward flexion, and curling the finger inwards through DIP and PIP flexion. The distal, middle, and proximal phalanges are also labelled in green to show some of the hand design parameters. b) Visualization of simulated top-performing hand design V7

during interactions with objects and the environment as well as robustness to error due to compliance [191]. Bauer et al. [7] show that crease-like deformations in soft robotic hands can be kinematically approximated as rigid joints for design validation. Similarly, we use DASH, a Dexterous Anthropomorphic Soft Hand [101] with joint-like creases that is customizable, low-cost, easy to rapidly prototype and evaluate in the real world using teleoperation.

### DASH Morphology

From previous work [101], we use five existing design iterations of DASH, named V1, V2, V3. V4, and V5 to start our design optimization process. Each version of DASH incorporates four fingers: the thumb, index, middle, and ring fingers. To ensure modularity, every finger, including the thumb, shares an identical design. Each of these fingers consists of three joints along its length, moving from the base towards the fingertip: the metacarpophalangeal (MCP) joint, proximal interphalangeal (PIP) joint, and distal interphalangeal (DIP) joint. The joints for each individual finger are illustrated in Figure 7.2.

Each finger has four tendons (as shown in Figure 7.2), two are situated along the sides of the MCP joint, near the palm, responsible for abduction and adduction—permitting the fingers to move both closer together and farther apart. These two tendons are attached to a bi-directional pulley actuated by a single motor, collectively referred to as the adduction-abduction tendon. The MCP flexion tendon is used for the forward folding of the finger at the MCP joint, perpendicular to the abduction-adduction tendon's motion. Lastly, the DIP PIP flexion tendon extends through the entire length of the finger in order to fully curl the finger inwards.

| ***Design Params*** | | *Min* | *Max* | v3 | v5 | v6* | v7* |
|---|---|---|---|---|---|---|---|
| Palm width | | 69 | 99 | 84 | 84 | 92 | 92 |
| Palm height | | 69 | 99 | 84 | 84 | 74 | 74 |
| Position | ff | (8, 64) | (48, 84) | (28, 84) | (28, 84) | (28, 84) | (29, 83) |
| | mf | (-20, 64) | (20, 84) | (0, 84) | (0, 84) | (0, 84) | (0, 84) |
| | rf | (-56, 64) | (-16, 84) | (-28, 84) | (-28, 84) | (-36, 84) | (-36, 83) |
| Orientation | ff | 0 | 45 | 0 | 0 | 0 | 2.9 |
| | mf | -35 | 35 | 0 | 0 | 0 | 0 |
| | rf | -45 | 0 | 0 | 0 | 0 | -2.9 |
| Proximal length | | 35 | 55 | 45 | 45 | 45 | 45 |
| Middle length | | 8 | 28 | 20 | 20 | 18 | 18 |
| Distal length | | 25 | 45 | 35 | 35 | 35 | 35 |

Table 7.1: Hand design parameter ranges tested in simulation to find the optimized robotic hand design where ff, mf, and rf refer to first (index) finger, middle finger, and ring finger, respectively. Unit is millimeter for lengths and degree for angles. Designs with * are optimized hand designs.

### Fabrication and Customization

In Figure 7.1(c), the fingers and palm were 3D-printed from Ninjaflex Edge (83A shore hardness) [96] using a FlashForge Creator Pro 2 printer. Tendon routing and finger morphology can be easily changed to 3D-print a new robotic hand design, facilitating rapid prototyping and evaluating. We use 12 Dynamixel XC330-M288-T motors [192] in the motor housing which is 3D-printed from PLA and shown below the palm in Figure 7.1(b). Our real robot experiments used the robot hand mounted to an xArm7 robotic arm [193].

### Design Parameters

While 3D-printing our soft dexterous hands allows us endless possibilities in terms of design, we constrain the design parameters to ones that allow for significant manipulation behavior changes. In prior work [101], the design changes among DASH v1 to v5 included parameters such as palm size, finger length, joint height and width, thumb arrangement, and fingertip shape.

For our simulated design optimization, we restrict the design changes to the following parameters illustrated in Figure 7.2a: palm size, finger links lengths (proximal, middle, distal phalanges' lengths), finger width, finger thickness, finger position, and finger orientation. The ranges of these parameter values are detailed in Table 7.1. Figure 7.2b shows the visualization of the same hand CAD shown in Figure 7.2a in simulation where the hand parameters match for both hands. However, features like finger surface geometry were not varied during optimization. Tendon arrangements and actuation were integrated into the design after determining the optimized hand design in simulation.

## 7.4 Design and Policy Co-Optimization

We assume the hand robot can be defined and represented by its design parameters. This assumption is true for the design space of our DASH hand where we fix the topology of kinematic connections of bodies and joints. In this section, we describe our proposed method for automatic optimization of the DASH hand design parameters.

### 7.4.1 Problem Definition and Preliminaries

We formulate the robotic manipulation task as a Markov Decision Process (MDP) specified by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \gamma)$, where $\mathcal{S} \subseteq \mathbb{R}^S$ is the state space, $\mathcal{A} \subseteq \mathbb{R}^A$ is the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition function, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1]$ is the discount factor. A policy $\pi : \mathcal{S} \to \mathcal{A}$ maps a state to an action where $\pi(a|s)$ is the probability of choosing action $a$ at state $s$. We assume that all possible robot designs share the same state space $\mathcal{S}$ and action space $\mathcal{A}$. We assume a robot can be defined by $D$ independent design parameters that only impact the transition dynamics $\mathcal{T}$. Suppose $\rho^{\pi,\theta} = \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$ is the episode discounted reward when a robot with design parameters $\theta \in \mathbb{R}^D$ executes policy $\pi$. The optimal policy $\pi^*_\theta = \arg\max_\pi \mathbb{E}[\rho^{\pi,\theta}]$ on robot $\theta$ is the one that maximizes the expected value of $\rho^{\pi,\theta}$.

Given the MDP for a certain manipulation task, the goal of our problem is to find the optimal design parameters of the robot design defined as

$$\theta^* = \arg\max_\theta \rho^{\pi^*_\theta, \theta} \tag{7.1}$$

This means robot designs should be compared by the performance of their respective well-trained expert policy. The difficulty of this optimization problem stems from two aspects. First, finding the expert policy $\pi^*_\theta$ by training each policy from scratch for each robot design $\theta$ is computationally expensive. Second, the robot design space is vast and has an exponential complexity with respect to the number of design parameters $D$.

To address these challenges in robot design optimization, we aim to improve policy optimization efficiency while allowing global search of robot design parameters. Recent advances in interpolation-based policy transfer [177, 178] inspire us to address this problem from the perspective of policy transfer. The intuition is that, if two robots are similar in their hardware configuration, their optimal policy should also be similar. Therefore, given a robot $\theta_{\text{source}} \in \mathbb{R}^D$ with an expert policy $\pi^*_{\theta_{\text{source}}}$ and another significantly different robot $\theta_{\text{target}} \in \mathbb{R}^D$, it may be possible to interpolate the two robots by producing a sequence of intermediate robots $\theta_1, \theta_2, \ldots, \theta_{K-1}, \theta_K$ that continuously changes from $\theta_1 = \theta_{\text{source}}$ to $\theta_K = \theta_{\text{target}}$. These intermediate robots act as the stepping stones for transferring the policy. It takes $K$ training phases to transfer the policy where in $i$-th phase, the policy is fine-tuned on $\theta_i$ until convergence. Since the difference between any two consecutive robots $\theta_i$ and $\theta_{i+1}$ in the sequence is sufficiently small, the overall training overhead for transferring of policy is much smaller than training the policy on $\theta_{\text{target}}$ from scratch.

### 7.4.2 Genetic Algorithm and Policy Transfer for Design Optimization

Our method leverages the above idea to efficiently transfer the expert policies on existing robot designs to multiple robot design candidates that are iteratively searched and trained.

Specifically, given the pool of $C$ existing robot designs $\mathcal{D}_e = \{\theta_i \in \mathbb{R}^D \mid i = 1, 2, \ldots, C\}$ and their respective well-trained expert policy $\pi^*_{\theta_i}$, we randomly generate a new robot design candidate $\theta_{\text{new}} \in \mathbb{R}^D$. To obtain the optimal policy on the new design candidate, it is natural to find the closest existing robot design with expert policy as the starting point for policy transfer to reduce the overall computational cost. Suppose $\theta_s = \arg\min_{\theta \in \mathcal{D}_e} ||\theta - \theta_{\text{new}}||$ is the robot with the smallest hardware difference to $\theta_{\text{new}}$ in $\mathcal{D}_e$. Then the next robot stepping stone can be obtained by moving the design parameter $\theta_s$ towards $\theta_{\text{new}}$. We employ a small step size $\xi \in \mathbb{R}^+$ for each step of design parameter change.

How can we randomly generate new hand design candidates? Given large $D$, it is computationally intractable to iterate all possible design candidates. Our intuition is that good robot designs are usually superior due to certain "traits", e.g. long fingers, wide motion ranges. It makes sense to share the good traits with other designs to search potentially better designs. Genetic algorithm is an optimization scheme that is capable of achieving this. It contains two parts: crossover and mutation. In every design exploration phase, we randomly sample two existing robot design candidates $\theta_1, \theta_2$ from $\mathcal{D}_e$. We perform element-wise random crossover on $\theta_1$ and $\theta_2$ to generate a new $\theta_{\text{new}}$. Then we mutate $\theta_{\text{new}}$ by adding a noise term that is uniformly sampled from $[-\theta_M, \theta_M]$, where $\theta_M \in \mathbb{R}^{+D}$. In this way, we ensure $\theta_{\text{new}}$ can inherit the good traits from existing good designs while allowing exploration in the design space for potential improvement over the current state of the art.

Note that the robot design pool $\mathcal{D}_e$ is not static. To fully re-use the previously searched good designs, apart from initial designs, $\mathcal{D}_e$ is dynamically populated with newly found designs as well, including the designs acting as policy transfer stepping stones. However, in order for the genetic algorithm to find better designs, $\mathcal{D}_e$ should only contain the most elite population of robot designs. Therefore, we set a threshold of robot policy performance $q$ for deciding whether a robot design is good enough to be added to the pool $\mathcal{D}_e$. Our overall design and policy co-optimization method is illustrated in Algorithm 2.

### 7.4.3 Remarks

Our robot design optimization scheme does not guarantee convergence as genetic algorithms are not guaranteed to converge to a global optima. However, with sufficient amount of computation, our method can cover enough space of possible good designs to result in optimized designs, which can serve as good candidates for real-world fabrication of our DASH hand.

Note that it is possible that in some iterations the randomly generated new design candidate $\theta_{\text{new}}$ is a bad one with low expected episode reward. In this case, during policy transfer, upon reaching a certain intermediate robot $\theta$ along the path, the well-trained expert policy $\pi^*_\theta$ will no longer reach the performance threshold $q$. Though the generated bad design will not provide new member for $\mathcal{D}_e$, it can still provide valuable information on what design parameter changes are not useful.

**Algorithm 2** Design and Policy Co-Optimization

**Notation Summary:** $\theta_i \in \mathbb{R}^D, i = 1, 2, \ldots, C$: design parameter of $i$-th candidate robot; $\pi^*_{\theta_i}$: expert policy on $i$-th candidate robot; $\xi \in \mathbb{R}^+$: evolution step size; $q \in \mathbb{R}$: reward threshold; $\mathcal{D}_e$: the set of robots with expert policy; $\theta_M \in \mathbb{R}^D$: element-wise mutation range. $\theta_U, \theta_L \in \mathbb{R}^D$: element-wise upper and lower bounds of design parameter.

$\mathcal{D}_e \leftarrow \{\theta_1, \theta_2, \ldots, \theta_C\}$;
**for** $i$ **in** $1, 2, \ldots, N$ **do**
  // sample new robot design candidate
  Sample $\theta_1, \theta_2 \sim \mathcal{D}_e$, mutation noise $n \sim U([-\theta_M, \theta_M])$;
  $\theta_{\text{new}} \leftarrow \texttt{random\_crossover}(\theta_1, \theta_2) + n$;
  $\theta_{\text{new}} \leftarrow \text{MAX}(\text{MIN}(\theta_{\text{new}}, \theta_U), \theta_L)$; // stay in bound
  // find closest source robot to transfer policy from
  $\theta_s \leftarrow \arg\min_{\theta \in \mathcal{D}_e} ||\theta - \theta_{\text{new}}||$;
  $\theta \leftarrow \theta_s$, $\pi^* \leftarrow \pi^*_{\theta_s}$;
  // transfer the policy by robot interpolation
  **while** $||\theta - \theta_{\text{new}}|| < \varepsilon$ **do**
    $\theta \leftarrow \theta + \xi \cdot (\theta_{\text{new}} - \theta)/||\theta_{\text{new}} - \theta||$;
    train expert policy $\pi^*_\theta \leftarrow \arg\max_\pi \mathbb{E}[\rho^{\pi,\theta}]$ by initializing policy with $\pi^*$;
    $\pi^* \leftarrow \pi^*_\theta$
    **if** $\mathbb{E}[\rho^{\pi^*_\theta,\theta}] > q$ **then**
      $\mathcal{D}_e \leftarrow \mathcal{D}_e \cup \{\theta\}$; // only keep elite robot candidates
**return** $\{(\theta, \pi^*_\theta) \mid \theta \in \mathcal{D}_e\}$;

## 7.5 Simulation Benchmark and Evaluation

### 7.5.1 Benchmark Definition

Using simulation to optimize the hand design requires designing a benchmark suite whose manipulation task and evaluation metrics can distinguish between similar hands. In this section, we propose a benchmark task suite for evaluating dexterous hands. In contrast to previous works that separate object relocation and in-hand reorientation [28], we combine object relocation and reorientation into one task where the goal is to pick up the object and then reorientate the object in-hand to desired relative 6D pose between hand and object. In this way, both object grasp ability and hand dexterity can be fully evaluated in a single task.

To evaluate the full potential of the hand designs, we include six objects with diverse geometry in our benchmark suite. The six objects include `barbell`, `board`, `cross3d`, `pen`, `ring` and `sphere` as illustrated in Figure 7.1. The objects are chosen such that the their geometry is diverse enough to evaluate different taxonomy of hand grasping and reorientation. For example, `sphere` can help evaluate spherical grasp while `board` focuses more on power grasp. Each object is instantiated with three different uniform scales: 0.75×, 1×, and 1.25×, resulting in 18 object instances in total. The hand will be evaluated on manipulation tasks using all 18 object instances.

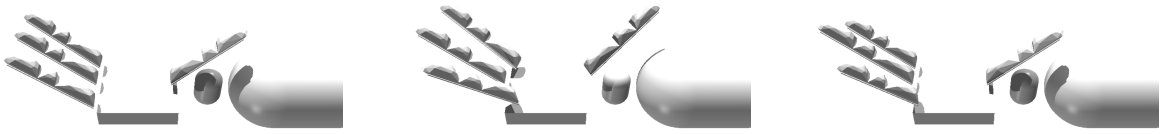


Figure 7.3: Top three optimized hands. From left to right, the success rate AUC (%) of the hands are 53.32%, 53.07% and 52.47% respectively.

### 7.5.2 Evaluation Metrics

Defining an episode success condition and using the success rate as the evaluation metric has been a popular approach in previous works. However, using a discrete signal of episode success or failure cannot sufficiently distinguish subtle difference in hand designs. On the other hand, defining dense and continuous evaluation metrics cannot easily generalize to other tasks and is not straightforward to define for our object relocation and reorientation task.

We propose a new evaluation metric to address the above problem. During evaluation, we apply an unknown external force $F$ to the object in simulation. The direction of the external force is random and stays the same throughout the episode. Due to unknown external force being applied, achieving high success rate is difficult. We then measure the average success rate under different external force magnitudes. Formally, let $S(\pi, \theta, F)$ be the average success rate of executing policy $\pi$ on robot $\theta$ with unknown external force of magnitude $F$. Then our evaluation metric of robot $\theta$ is defined as

$$M(\theta) = \frac{1}{F_{\text{max}}} \int_0^{F_{\text{max}}} S(\pi_\theta^*, \theta, F) dF \tag{7.2}$$

where $F_{\text{max}} = 1\text{N}$. $M(\theta)$ is essentially the area-under-curve (AUC) of success rate $-$ external force curve. In our experiments, we found that $M(\theta)$ is a more distinguishable metric than naively using success rate $S(\pi_\theta^*, \theta, 0)$. In practice, the integral in Equation (7.2) is approximated by discretized summation.

### 7.5.3 Other Implementation Details

To evaluate a large number of hand designs on a variety of tasks, it is imperative to perform large-scale simulation. We use Issac Gym [194], a GPU-empowered parallel simulator as the physics simulation engine. For each object instance, we launch 256 simulation environments, resulting in 4,608 environments running in parallel.

Given 18 different object instances, a naive solution is to train 18 independent policies for each object instance. This solution does not fully exploit the potential of parallelism of GPU. Instead, we train a single control policy to manipulate all 18 object instances. The policy is conditioned on the object ID, so that one policy can control the manipulation of all 18 object instances. The object ID is represented as a one-hot vector and sent into the policy neural network together with original state vector. We illustrate three of the top searched hands in Figure 7.3.

## 7.6 Real-World Design Evaluation

A subset of the simulated hand designs were tested in the real-world using teleoperation. All six objects, `barbell`, `board`, `cross3d`, `pen`, `ring` and `sphere`, were tested in the real-world at a single scale ($1\times$). While $256$ randomized goal poses were tested in simulation, we limit our real-world robotic hand design evaluation to 6 goal poses from randomized initial positions for the pick up and reorient manipulation task. We fabricated two optimized hands from simulation from the top 25 hands, where the first ranked hand is the leftmost design shown in Figure 7.3. The tested hands include DASH V3, DASH V5, the 18th best ranking hand, and the 1st ranking hand in simulation. We refer to the 18th and 1st best ranking simulation hands as V6 and V7, respectively.

The hand design parameter values of V3, V5, V6, and V7 are listed in Table 7.1. DASH V3 and V5 are from previous work [101] where V3 has the thumb arranged directly opposing the index finger and V5 has the thumb rotated by $22.5°$ towards the ring finger. Both of these hands have the thumb placed in the bottom right corner of the palm, much like the human hand. The optimized hands, V6 and V7, have thumbs in the bottom middle of the palm, as shown in Figure 7.2. These hand designs have a wider but shorter palm with the ring finger offset farther away from the middle finger than the index finger. Unlike V6, V7 also has the index and ring finger turned toward the middle finger by $3°$. While our optimization did not include fingertip shape, V3 and V5 had a wedge-like and flat fingerpad-like fingertip, respectively. Meanwhile, V6 and V7 had rounded spherical fingertip shapes (as shown in Figure 7.1b).

### Experiment Setup

Previous work [101] has shown that teleoperation removes the necessity to learn control policies for new soft hand designs for quick evaluations of the hand's capabilities in real-world experiments. We use Manus [98] VR gloves to teleoperate a 7-DOF robotic arm and the fabricated soft hand for various manipulation tasks to evaluate the hand designs in the real-world. Each of the hand designs is calibrated individually to map joint-to-joint control of DASH as well as scaling motions such as adduction-abduction motions to be exaggerated from human hand motion for easier teleoperation. Further calibration and teleoperation implementation details can be found in previous work [101].

Our real-world teleoperation evaluation setup involves two people. First, we use an experienced teleoperator to perform manipulation tasks with each tested hand for the 6 objects and repeat each goal pose three times. Once a goal pose is achieved, the second person will interactively score the goal pose on a scale of 0 to 1 based on grasp stability. Scores are decided by both people and discussed upon disagreement. These qualitative metrics are similar to related work that interactively tested grasp quality [190]:

- $1$ - stable grasp (unmovable by small disturbance force)
- $0.75$ - object moves by disturbance force but will not drop
- $0.5$ - object can be moved and dropped by disturbance force
- $0.25$ - grasp fragile and will not be able to carry object
- $0$ - grasping or achieving goal pose failed

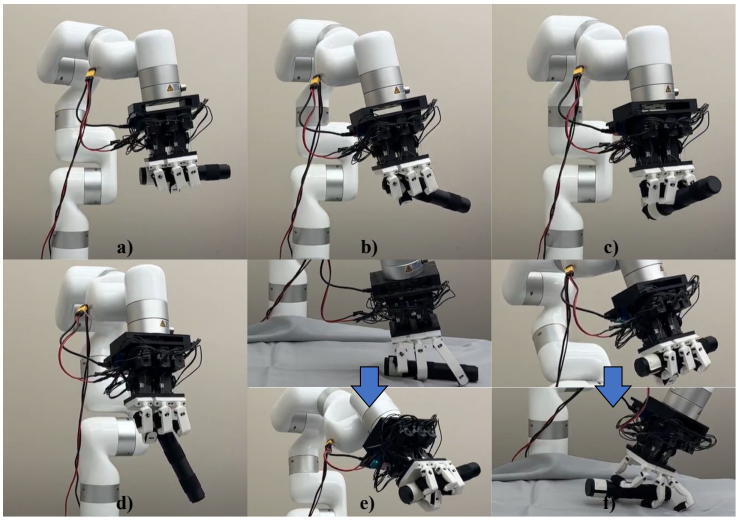



Figure 7.4: The 6 goal poses (shown for `pen` object) used for real world teleoperated manipulation tasks.

The teleoperator tested v3, v5, v6, and v7 sequentially in one order and repeated each hand in the reverse order to remove the effects of learning through experience teleoperating DASH for these tasks. Each hand was tested twice on each object and scores are averaged. The average and maximum scores across the two repetitions of each hand are reported in Tables 7.2- 7.3. The objects are augmented with grip tape and foam tape in order to increase friction and gripability of the smooth 3D-printed objects. White tape is used to mark the orientation of the object due to symmetry, as shown in Figure 7.4. For each object, the six goal poses are referred to as horizontal (Figure 7.4a), tilt $45\,\mathrm{deg}$ (Figure 7.4b), cigarette grasp (Figure 7.4c), vertical (Figure 7.4d), table to precision grasp to power grasp (Figure 7.4e), and power grasp to precision grasp to table (Figure 7.4f).

These goal poses are shown for the object `pen` in Figure 7.4. The last goal pose starts with the object in-hand in a power grasp and requires the object to be gently placed on the table. The quality metrics for goal pose 6 uses a different scoring metric:

- $1$ - graceful placement of object on table within 3cm drop height
- $0.5$ - object initially contacts table and falls greater than 3cm
- $0$ - object falls from hand greater than 3cm height with no initial contact with table

**Teleoperation Evaluation Results**

As shown in Table 7.2, the averages and maximum scores for each object show that on average the optimized hands v6 and v7 performed the best. From Table 7.3, we see that v7 struggled with pose 6 the most compared to other goal poses and hand designs. With two fingers rotated inwards towards the middle finger, we observe that v7 can cause unexpected rotations of the object during manipulation. This made v7 easier for power grasps rather than precision grasps. `cross3d` is our smallest tested object and required precision, which v5 performed the best on.

v5 has a larger palm and longer fingers than the optimized hands, which performed best on

| ***Objects*** | v3 | | v5 | | v6 | | v7 | |
|---|---|---|---|---|---|---|---|---|
| | avg | max | avg | max | avg | max | avg | max |
| barbell | 0.81 | 0.86 | 0.92 | **0.99** | **0.95** | 0.96 | **0.95** | **0.99** |
| board | 0.78 | 0.83 | 0.94 | 0.99 | **0.97** | **1.00** | 0.92 | 0.99 |
| pen | 0.78 | 0.79 | 0.79 | 0.89 | **0.92** | **0.96** | 0.91 | 0.94 |
| ring | 0.85 | 0.94 | 0.83 | 0.85 | **0.92** | **0.96** | 0.90 | 0.90 |
| cross3d | 0.87 | 0.90 | 0.88 | **0.99** | 0.88 | 0.93 | **0.94** | 0.94 |
| sphere | 0.67 | 0.78 | 0.83 | 0.92 | **0.96** | **1.00** | 0.83 | 0.85 |
| avg | 0.79 | 0.85 | 0.87 | 0.94 | **0.93** | **0.97** | 0.91 | 0.94 |

Table 7.2: Teleoperation evaluation results for pick up and reorient tasks for each of the six objects on all goal poses using different hand designs. The above table shows both the average and max grasp quality score achieved.

| ***Goal Pose*** | v3 | | v5 | | v6 | | v7 | |
|---|---|---|---|---|---|---|---|---|
| | avg | max | avg | max | avg | max | avg | max |
| pose 1 | 0.86 | 0.93 | 0.94 | **1.00** | **0.98** | 0.99 | 0.97 | 0.99 |
| pose 2 | 0.80 | 0.88 | **0.92** | **0.99** | 0.91 | 0.92 | 0.88 | 0.94 |
| pose 3 | 0.85 | 0.86 | 0.80 | 0.89 | 0.89 | **0.99** | **0.94** | 0.96 |
| pose 4 | 0.80 | 0.86 | 0.79 | 0.81 | 0.91 | **0.96** | **0.93** | 0.94 |
| pose 5 | 0.73 | 0.83 | 0.92 | 0.99 | 0.95 | **1.00** | **0.97** | 0.99 |
| pose 6 | 0.72 | 0.78 | 0.81 | 0.94 | **0.96** | **1.00** | 0.76 | 0.78 |
| avg | 0.79 | 0.86 | 0.86 | 0.94 | **0.93** | **0.98** | 0.91 | 0.93 |

Table 7.3: Teleoperation evaluation results for pick up and reorient tasks for each of the goal poses on all six objects using different hand designs. The above table shows both the average and max grasp quality score achieved.

the barbell and board objects as well as outperforming v6 on the tilt goal pose 2. Since goal pose 2 requires the object to be at a 45 degree angle, having the thumb on the bottom right corner of the palm was helpful for v5. Between v6 and v7, they both performed similarly on barbell and board. v6 did the best on pen and ring, and v5 did the best on cross3d. However, v5, v6, and v7 excelled similarly at the power grasp from the table (goal pose 5).

Across objects and goal poses, v3 scored lower than other hands. We observed that the combined effect of not being able to fully curl the fingers and its wedge-like fingertips caused v3 to have more unstable grasps compared to the other hands. In previous work [101], finger curling ability was improved for v5, so we see that the scores for v5 are better than v3 for each goal pose in Table 7.3. The easiest goal pose for all hands was the horizontal goal pose 1 and the most difficult pose was starting from power grasp and placing the object on the table using a precision grasp.

| ***Simulation*** | v3 | v5 | v6 | v7 | v7-*wedge* |
|---|---|---|---|---|---|
| barbell | 0.28 | **0.37** | 0.30 | 0.32 | 0.23 |
| board | 0.87 | 0.90 | **0.92** | 0.90 | 0.89 |
| pen | 0.72 | 0.72 | **0.74** | 0.79 | 0.67 |
| ring | 0.09 | 0.09 | 0.09 | 0.08 | 0.09 |
| cross3d | 0.06 | 0.06 | **0.07** | **0.07** | **0.07** |
| sphere | 0.54 | 0.35 | 0.53 | **0.60** | 0.42 |
| avg | 0.43 | 0.42 | 0.44 | **0.46** | 0.40 |

Table 7.4: Simulation evaluation results for pick up and reorient tasks for each of the six objects at scale $1\times$ on randomized goal poses using different hand designs. v7-wedge is v7 with a wedge-like fingertip shape. The above table shows AUC of success rate as given by equation (7.2).

### Simulation vs. Real Results

If we look at the performance by object, simulation and teleoperation results follow a similar trend. v6 performs best on most objects in simulation including board, pen, and cross3d as shown in Table 7.4. Simulation results also show that v3 and v5 tied for best score on sphere, and v5 performed best on barbell. While real-world experiments showed comparable scores on all six objects, simulation results have a high variation depending on objects. ring and cross3d have the lowest scores and board and pen have the highest scores. In teleoperation experiments, pen and sphere have the lowest scores and barbell and board have the highest scores, but by a small margin. This correlates with our observation that the compliance of our soft hands helped most with ring and cross3d objects for conforming to the object for stable grasps. Overall, the performance of hand designs simulated as rigid bodies correlate with the performance of our teleoperation evaluation results with v6 performing best on most test objects.

### Preliminary Evaluation with Varying Fingertip Shape

We also tested v7 with a different fingertip shape in simulation and using teleoperation. As shown in Table 7.4, v7 with a rounded fingertip shape generally performed better than v7 with a wedge-like fingertip shape. When tested on the real robot, we found that the wedge-like fingertip for board, cross3d, sphere scored $0.89$, $0.99$, and $0.96$, respectively. This shows that fingertip shape results do not follow the trend of simulation results. Teleoperation results showed that v7 with a wedge-like fingertip performed better than or as good as v7 with rounded fingertips for cross3d and sphere. We observed that v7's difficulty with precision was remedied with thinner fingertips but made stability worse for board.

### Preliminary Evaluation with other Teleoperators

In addition to our experienced teleoperator experiments shown in Tables 7.2-7.3, preliminary results with two non-expert teleoperators on v3, v5, and v7 corroborate our findings. The two

operators tested only the `board` object on the three hands in a random order and repeated the reverse order. We then aggregated across repetitions and found that V7 performed best based on maximum scores and V5 performed best based on averaged scores. This is similar to Table 7.2 finding that showed similar performance with V5 and V7 on board.

## 7.7 Discussion & Conclusion

Using genetic algorithms and policy transfer, we were able to optimize on the design of a 16-DoF anthropomorphic soft robotic hand. While simulation used trained expert policies to manipulate the objects, teleoperated experiments used potentially suboptimal policies to achieve the goal pose. This is a limitation of our evaluation as teleoperators are not able to find the best policy for each hand which can artificially assign lower scores to a design. We compensated for this limitation by allowing the teleoperator to practice each task for every goal pose and object, allowing them to learn a near-optimal policy. Interactive grasp quality testing during teleoperation experiments provided finer insights into the limitations and advantages of each hand design. Using only task success in both simulation and real-world experiments would have made it difficult to differentiate between hand designs.

Despite the sim-to-real gap arising from modeling the hand as a rigid body, we saw the same trends in performance across both simulation and real world evaluations, with V6 scoring higher on most objects. Both optimized designs V6 and V7 outperformed existing designs, V3 and V5 from prior work [101], which were a result of manual design iteration in the real-world. While smaller scale features like finger geometry were not optimized, larger scale features like finger arrangement and palm size were successfully guided by simulation results. In contrast, features like fingertip shape did not show positive correlation across simulation and real world. A potential future direction would be to investigate the design parameters that are suitable for design optimization in simulation. Another direction for future work involves directly deploying learned policies to evaluate hand designs in the real world to mitigate the limitations of teleoperation.

# Chapter 8

# Evaluating Automated Hand Design on DASH-30 Tasks

## 8.1 Automated hand design performance

In Chapter 7, we automated design iteration of DASH and found an optimized design, V7 using simulation. The tasks we optimized for in simulation as well as tested in the real-world include picking up and reorienting six objects in-hand (barbell, ring, board, cross3D, pen, and sphere). While these tasks test dexterity and manipulator capabilities, we wish to see how well the optimized hand generalizes to tasks we did not evaluate during optimization. In Chapter 5, we manually designed and iterated on five versions of DASH using an human-in-the-loop approach and the DASH-30 tasks for evaluation. In this chapter, we aim to answer two questions, 1) how well does V7 generalize beyond the tasks used during optimization, and 2) does our optimized design V7 perform better than our manually iterated hand designs V1 to V5. To answer these questions, we focus on the DASH-30 tasks introduced in Chapter 5.

## 8.2 DASH-30 Tasks

In Chapter 7, Table 7.3 and Table 7.2 show that on most tasks, V7 scores similarly or better than V5, and V7 outperforms V3. These results are based on approximately 324 teleoperation experiments to show real-world hand design performance on 6 objects and 6 goal poses per object (each goal pose repeated three times). To further understand the capabilities and limitations of V7, we evaluate V7 on DASH-30 tasks. Thus, we perform 150 teleoperation experiments to compare against the V1 through V5 data (including Allegro dexterous hand results as baseline) discussed in Chapter 5.

### 8.2.1 DASH V7 results on DASH-30 tasks

DASH V7 has a smaller palm, round fingertips, a centered thumb, a gap between the middle and index fingers, and an index and ring finger rotated in towards the middle finger. This hand design performs better than Allegro and previous iterations of DASH, as shown in Figure 8.1,

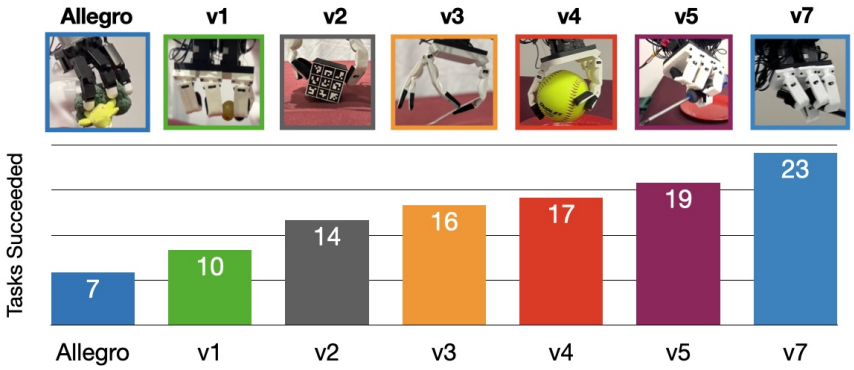



Figure 8.1: Tasks succeeded (out of 30 total) for V1, V2, V3, V4, V5, and V7, where the task succeeded for all five repetitions of the task. Each hand design iteration improved with the automated design iteration using simulation, V7 outperforming the previous designs and our baseline Allegro hand. Each hand did best on different tasks, as shown in the insets above the graph.

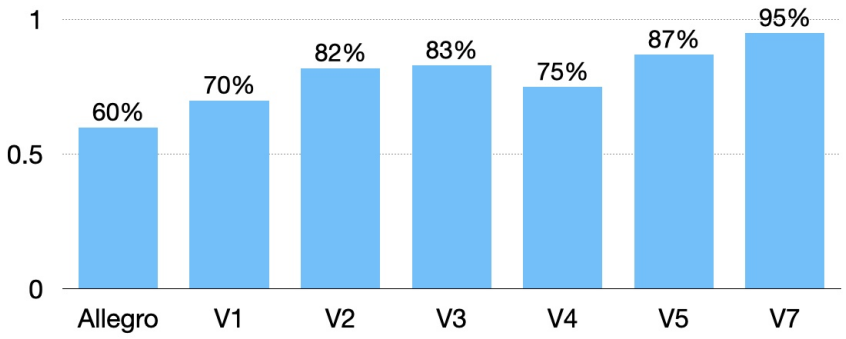



Figure 8.2: Task success on DASH-30 suite of tasks for 6 iterations of DASH including the baseline Allegro hand. Our automated hand design iteration, V7, improves on our manually designed iterations V1 to V5 and outperforms Allegro.

according to the number of tasks where all five out of five repetitions were successful. Allegro, V1, V2, V3, V4, V5, and V7 succeeded on all five repetitions of 7, 10, 14, 16, 17, 19, and 23 tasks, respectively. The increase of 4 tasks between V1 and V2 is similar to that between V5 and V7. Based on overall task success (over all 150 experiments per hand), Figure 8.3 shows that V7 succeeded on 95% of tasks, which is higher than the other design iterations.

Similar to the task specialization we saw with V1 to V5, V7 did best on the Pen object, as shown in the top right of Figure 8.1. The task performance of all hands including V7 is shown in Figure 8.3. V7 succeeded on all five repetitions of all tasks except for seven tasks. The seven tasks are Egg, Chip, M&M, Card Pickup, Dice Rotation, Grape, Plastic Bag. V7 succeeded on four out of five repetitions for all of these tasks except for twisting the grape off of the stem, which was successful for three repetitions of the task. We represent these tasks as categories from Table 5.2 in Figure 8.4, where V7 does better or the same as previous iterations except for the Twist category.

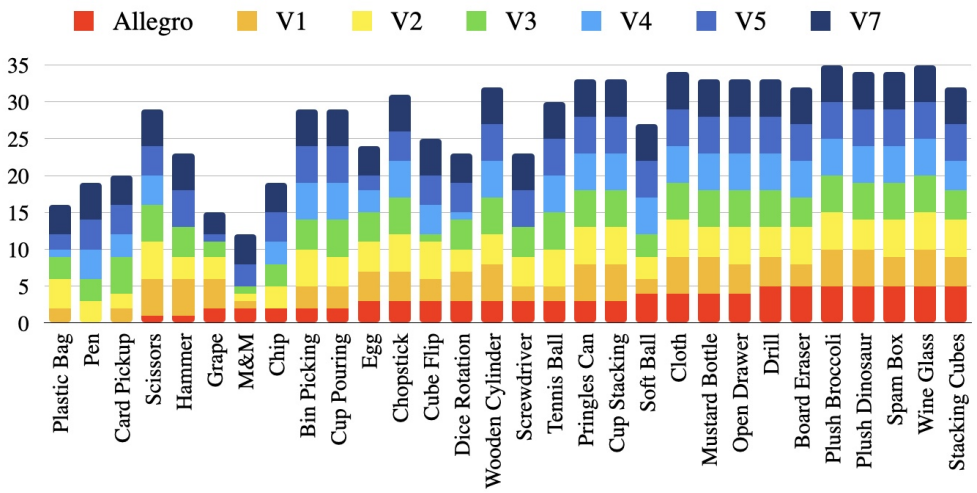



Figure 8.3: Task performance over 6 iterations of DASH on all DASH-30 tasks. The tasks are ordered difficult to easy from left to right, according to task performance of Allegro.

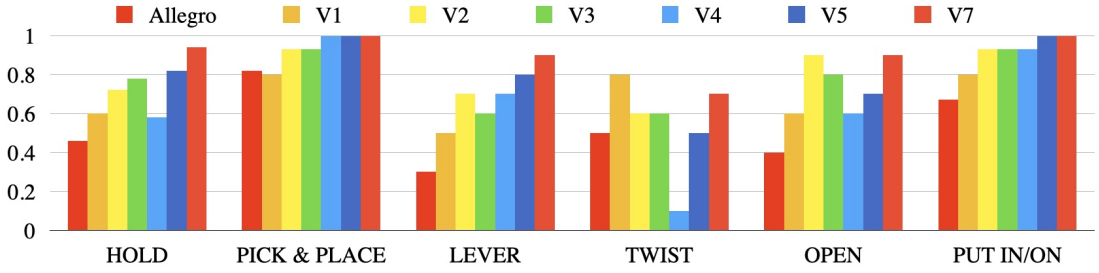



Figure 8.4: Task performance across DASH V1, V2, V3, V4, V5, V7, and Allegro as baseline on each category of tasks from Table 5.2.

There are two tasks where V7 performs better than all previous design iterations. Figure 8.5 shows tasks with hand performance differences across iterations, including V7. V7 does better than V1 to V5 on Pen and M&M objects. These are both small objects that require precision. The smaller palm and round fingertips were useful for these tasks. On the other hand, V7 performed worse or the same as previous iterations of DASH (V1 and V2) for Grape. This is interesting as V1, V2, and V7 all have the same fingertip shape that might have had a disadvantage toward having more contact with the grape while twisting it off of the stem.

### 8.2.2 DASH-V3 results repeated

Due to the learning effects we observed in Chapter 7, we wanted to understand how a previously evaluated hand would perform after evaluating V7. The results of the DASH-30 tasks for repeating V3 as well as previously evaluated scores are shown in Table 8.1. Overall, V3 previously succeeded 124 tasks out of 150 and V3-repeat succeeded on 123 tasks. However, V3 succeeded on all five repetitions of 18 tasks which is two more than previously observed and failed on 2 tasks (zero successes) which is two more than before. Some of differences from the previ-

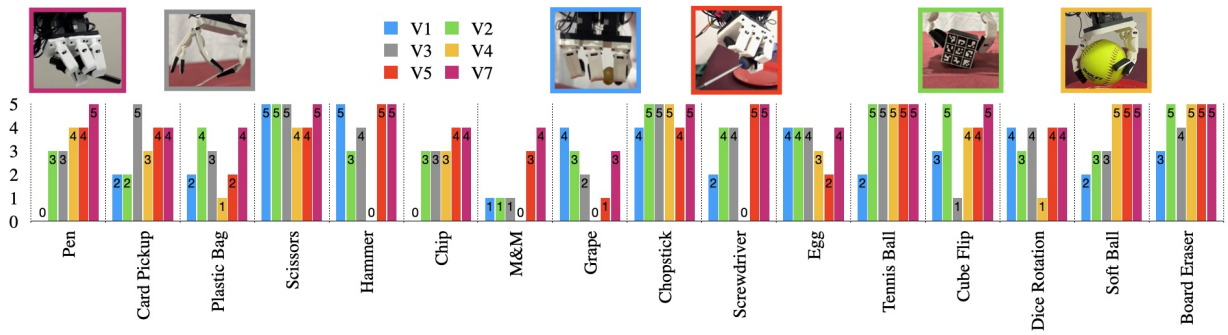



Figure 8.5: Subset of tasks with different performance across DASH V1 to V7 on specific tasks. The top row of inset images shows representative tasks of successful tasks for each hand.

ous evaluation of V3 include Hammer (improved from 4 repetitions to 5 repetitions) and Cube Stacking (decreased from 5 repetitions to 4 repetitions). Bigger changes in performance include performing worse on small objects such as Chopstick (dropped from 5 to 2 reps), Pen (dropped from 3 to 0 reps), M&M (dropped from 1 to 0 reps) due to the difficulty in accomplishing these tasks in comparison to V7's stability with thinner objects. Learning effects is an open challenge for teleoperation evaluation but we see that repeating V3 evaluation showed relatively the same performance for DASH-30 tasks.

## 8.3 Discussion

Our experimental results evaluating DASH V7 on the DASH-30 tasks shows that V7 outperforms V1 to V5 on tasks that were not seen during optimization, thus generalizing to tasks involving capabilities such as lever, twist, open, etc. During automated design iteration of V7, we only tested and evaluated tasks involving picking up and reorienting objects such as pens, rings, spheres, etc. However, our optimization and evaluation criteria using external disturbances resulted in a hand design that favored stable and robust grasps, which corresponds to our high performance results on DASH-30 with V7. Thus, we show that our optimized hand design V7 performs well on a different set of tasks, DASH-30, and performs better than our manually iterated hand designs V1 to V5 on these tasks.

| ***Task Success*** | *Allegro* | v1 | v2 | v3 | v4 | v5 | v7 | v3-repeat |
|---|---|---|---|---|---|---|---|---|
| $5/5$ | 7 | 10 | 14 | 16 | 17 | 19 | 23 | 18 |
| $4/5$ | 11 | 18 | 22 | 23 | 21 | 26 | 29 | 23 |
| $3/5$ | 20 | 22 | 28 | 27 | 24 | 27 | 30 | 26 |
| $2/5$ | 25 | 27 | 29 | 28 | 24 | 29 | 30 | 28 |
| $1/5$ | 27 | 28 | 30 | 30 | 26 | 30 | 30 | 28 |
| $0/5$ | 3 | 2 | 0 | 0 | 4 | 0 | 0 | 2 |

Table 8.1: Results of experiments. Task success measures the cumulative number of tasks where the hand succeeded. For example, $4/5$ means at least four out of five repetitions of the task were successfully completed and $0/5$ means none of the repetitions were successful.

# Chapter 9

# Discussion

In Chapters 3- 4, we looked at the design and evaluation of compliant 3-DOF delta manipulators. Chapters 5- 6 design and evaluate DASH, a tendon-driven anthropomorphic 16-DOF soft hand. After we've looked at the design iteration and evaluation of these two different manipulators, we investigated an automated design iteration approach in Chapter 7 to see if simulated hand designs could inform soft robot hand design in the real world. We find that our top simulated hand designs perform well in the real world, and as discussed in Chapter 8, our simulation optimized hand design outperforms our manually iterated hand designs.

## 9.1 Research Questions

After investing design iteration of compliant manipulators in manual and automated design processes as well as simulated and real-world environments, we distill our findings into four research questions:

1. What design parameters affect dexterity in multi-fingered hands?
2. What tasks can guide the design iteration of dexterous hands?
3. Can rapid prototyping and teleoperation evaluation enable design iteration in the real world?
4. How can simulating hand designs help with design iteration?

### 9.1.1 Design Parameters

**Increasing dexterity does not necessarily require adding motors or fingers.**

We investigated design parameters for DASH while automated design iteration using simulation and manual design iteration in the real world. For each of these scenarios, we focused on different sets of design parameters and evaluated how the resulting designs performed on dexterous manipulation tasks. Thus, we can heuristically show which design parameters affected dexterity in our implementation of 16-DOF dexterous anthropomorphic tendon-driven soft hands.

First and foremost, all of our DASH hand designs have the same number of fingers, degrees of freedom, modularity of the fingers (all fingers are identical to each other), fabrication tech-

nique, material properties, actuation, motor specifications, and number of motors. Increasing dexterity in a multi-fingered hand does not necessarily require adding more motors or fingers in the hand. Our findings indicate that enhanced dexterity and capabilities can be achieved without such additions in our DASH setup.

**Advancing dexterity requires attention to hand kinematics and compliance.**

In Chapter 5, the hand design parameters that change between V1 and V5 are palm size, finger length, MCP diameter, MCP height, DIP crease width, thumb angle, fingertip edge, and fingertip thickness (as shown in Table 5.3), where five dash iterations are designed manually using teleoperation evaluation in the real world. In Chapter 7, the hand design parameters that were changed across automated design space search included palm width, palm height, finger position, finger orientation, proximal phalanges' height, middle phalanges' length, and distal phalanges' length. Our automated design iteration hand parameters are a subset of the ones we searched in manual design iteration in the real world. Parameters such as MCP diameter and MCP height affected the finger stiffness at the MCP joint which affected the finger's ability to fold into the palm. For example, the stiffness was too high for V4 which resulted in poor performance on tasks like picking up the Screwdriver object. Our automated design iteration parameters did not include finger geometry which was covered by MCP diameter, MCP height, DIP crease width, fingertip edge, and fingertip thickness. DIP crease width resulted in different finger curling abilities, where V5 had the largest DIP crease width and the best curling ability out of our DASH design iterations. Similarly, fingertip shape was affected by fingertip edge and fingertip thickness parameters, where we varied the shape from rounded fingertips to flatter wedge-like fingertip shapes. Hands like V3 benefitted from flatter fingertips on tasks like picking up a single card from a deck of cards.

Our manual design iteration led to various hand designs from V1 to V5. While we have made fine steps within the search space with our manual design iteration, we could have made coarser steps by overcorrecting more between iterations to understand larger tradeoffs in hand designs (i.e. anthropomorphic vs. claw-like hand structures). Some tradeoffs such as good power grasps with V2 and good precision grasps with V3 resulted in the average of the two thumb placements for V5. Our findings focused on purely arranging fingers and changing finger geometry in a way that makes the tasks easier to perform with a specific hand design. However, if we had a morphology degree of freedom, we could attempt something like a moving thumb placement for specific tasks.

Overall, we can say that our manual design iteration investigated palm size, finger arrangement, finger length, and finger geometry, while our automated design iteration tested the effects of palm size, finger arrangement, and finger length. The key deficit of our automated design iteration design parameters compared to our manual design iteration process includes the lack of finger geometry tested in simulation. Furthermore, in Section 7.6, we show that simulated results and real-world results on testing fingertip shape did not correlate with each other. Thus, we leave it to future work to adjust simulated results to accurately model object interactions and hand contact models for automated hand design iteration involving parameters such as finger geometry.

**The three main categories of hand design parameters we tested are palm size, finger length, and finger arrangement.**

Even within the three main categories of hand design parameters, we tested in simulation and the real world, we have some insights into how to think about these design parameters and what their effects look like. While smaller palm sizes generally fared better on both our DASH-30 and pickup and reorient in-hand tasks tested in simulation, we found that palm width is not as important as palm height. Palm height dictates the distance between the thumb and non-thumb fingers. Shortening this distance and lengthening the fingers creates more overlap in the fingers' overlapping workspace for better fingertip reachability and increases dexterous manipulation skills. We especially saw this with the comparison of V3 and V5 compared to V6 and V7. Similarly, finger length is around the same for all of our DASH iterations with the earliest versions having the shortest fingers and worst finger reachability and opposability. However, our design automation process shortened the middle phalanges' length for our optimized hands V6 and V7. This is similar to the finger curling process that involves increasing the DIP crease width and decreasing the length of the middle phalanges. We found that to be optimal for our tendon-driven soft hand finger sizes. Lastly, finger arrangement mainly impacts thumb opposability and the robustness of precision grasps. Thumb opposability is most important for reorientation tasks that require opposing forces to stably rotate objects in hand, where the best opposability came from the centered thumb design in optimized hand designs V6 and V7. This allowed for the optimized hands to perform best on pick-up and reorientation tasks in Chapter 7. While the finger spread was tested coarsely in manual design iteration in the real world, we found that all hands except for V1 had tight packing of non-thumb fingers with different thumb placements. This results in a 22.5-degree thumb orientation placed at the bottom right-hand corner of the palm for our final design iteration in the real world. In our automated design search, we found the best design to have a centered thumb along the bottom of the palm, with a larger gap between the middle and ring finger than between the middle and index finger. However, the middle finger and thumb were directly opposite each other with no rotation of the thumb for both V6 and V7. This resulted in great thumb opposability and reachability to the other fingers (due to shorter palm height).

A natural question is are we converging towards human hands during optimization and how do our DASH hand designs compare to human hand morphology. After Chapter 7, we tested more hands, almost doubling the number of searched hands from around 400 to 821 hands searched. The top 25 resulting hands are shown in Figure 9.1, where V7 is the top 23rd hand (score 53.32). As we can see, V7 is one of the few hand designs that seem human-like where most of the hands lack symmetry or alignment of fingers along the edges of the palm. Our DASH hands compare similarly to human hands in terms of palm size but vary when it comes to phalanx proportions. Table 9.1 compares the DASH design parameters to human hand measurements where hand length measures the distance between the crease below the palm to the tip of the middle finger. We find palm measurements [5] and hand index (palm width divided by hand length as a percentage) to be similar between our hand designs and human hand proportions. While proximal phalanx lengths are similar, human hands have longer middle phalanges than distal phalanges [4] while DASH has shorter middle phalanges. This trend shows that our hand designs do not converge towards human-like hands and also show divergence from human hand finger arrangement with V7's thumb centered on the palm directly across from the middle finger.

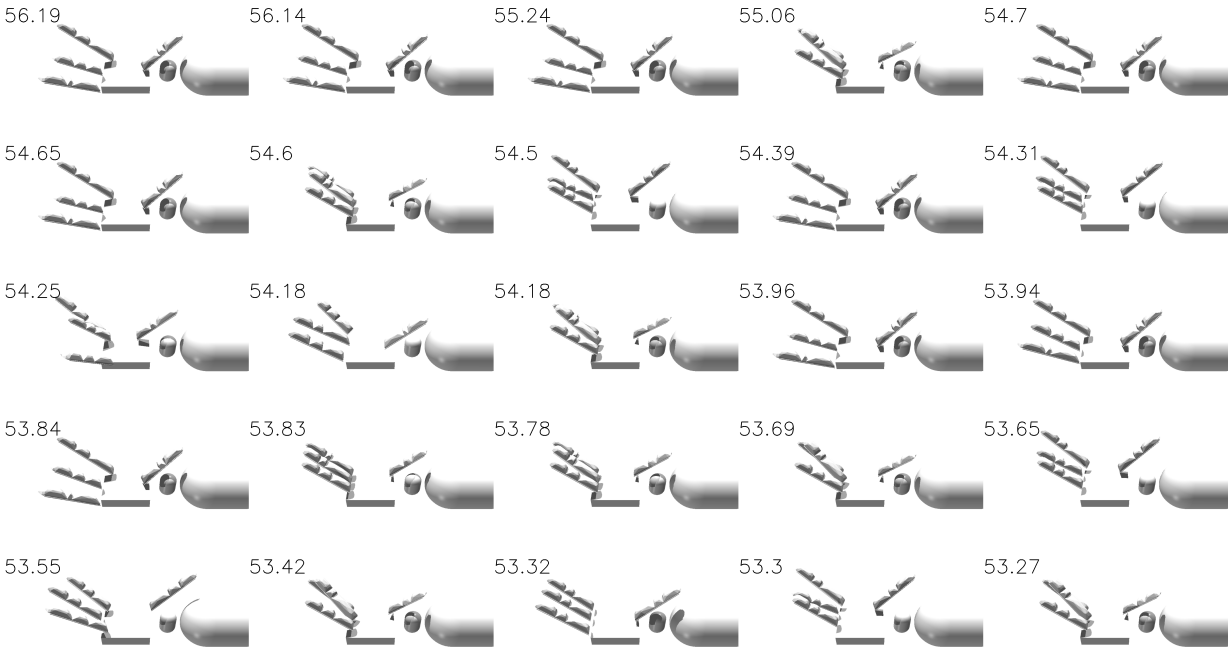



Figure 9.1: Top 25 optimized hands among 821 searched hands using design automation pipeline in Chapter 7.

| ***Human Hand Comparison*** | v1 | v2 | v3 | v4 | v5 | v6 | v7 | *Human* |
|---|---|---|---|---|---|---|---|---|
| Palm width (w) | 102 | 84 | 84 | 84 | 84 | 92 | 92 | 90 |
| Hand length (l) | 186 | 208 | 208 | 208 | 208 | 204 | 204 | 193 |
| Hand index (w/l*100) | 55 | 40 | 40 | 40 | 40 | 45 | 45 | 46 |
| Proximal length | 45 | 45 | 45 | 45 | 45 | 45 | 45 | 45 |
| Middle length | 20 | 20 | 20 | 20 | 20 | 18 | 18 | 26 |
| Distal length | 30 | 35 | 35 | 35 | 35 | 35 | 35 | 21 |

Table 9.1: DASH hand design parameters compared to human hand proportions, where proximal, middle, and distal phalanx lengths were taken from middle finger measurements [4] and hand index is the palm width [5] divided by hand length as a percentage. All measurements are in millimeters.

### 9.1.2 Choosing Tasks

Testing numerous hand designs requires suites of tasks or benchmarks that can properly evaluate these designs. The tasks should provide enough signal to distinguish hand designs in terms of task performance, which depends on the types of tasks we choose and the way we evaluate the task performance.

**Choosing a variety of tasks and objects allows differentiation in hand design performance.**

Our first suite of tasks includes DASH-30, which we hand-curated. DASH-30 includes 30 different objects and tasks. The tasks are broken down into categories, shown in Figure 9.2, with an emphasis on manipulation skills, i.e. hold, pick-and-place, lever, twist, open, and put in/on. The tasks vary in size, weight, and material which helped us evaluate and differentiate hand designs. Objects such as Chopstick, Pen, Egg, Chip, M&M, Card, and Grape are small and require fine precise motions of the fingers. Large objects include Wine Glass, Pringles Can, Mustard Bottle, Cup, Tennis Ball and Spam Box. Some objects are large and heavy such as the Drill, Hammer, Soft Ball, and Cube which require strong grip. The pouring, stacking, opening drawer, and Pringles Can reorientation tasks required strong enough grasps that moving the arm to finish the task post-grasp would not disturb the grasp. This requires both precision from the arm control to complete the task and the hand control during execution. Deformable objects such as Plush Broccoli, Plush Dinosaur, and Cloth were some of the easiest objects to grasp, even for our baseline hand Allegro. These tasks along with the lighter large objects (such as Wine Glass) were not informative tasks in terms of differentiating hand performance across iterations. As shown in Figure 9.2, most capabilities were retained through DASH iterations, except for precision tasks involving small objects. The tasks even v7 was not able to master included Egg, Chip, M&M, Card Pickup, Dice Rotation, Grape, and Plastic Bag. These were the most difficult tasks that require fine precision manipulation that would benefit from sensing in order to execute the tasks reliably. All of our experiments were done through teleoperation using solely visual feedback which was insufficient for such fine motor skills.

Most of the 30 tasks fall into the hold and pick-and-place categories since grasping different types of objects (i.e. rigid vs soft) can result in different grasping strategies. However, if we were to expand on the DASH-30 suite of tasks, adding more objects or tasks into each of the categories would give a more well-rounded picture of manipulation capabilities for hand designs. For instance, each category in Figure 8.4 would be based on the hand's performance on the same number of tasks. While the DASH-30 tasks span a large variety of objects, object types, and task setups, some of the tasks such as open and put in/on rely on grasping and reliable arm motion. This can be comparable to hold and pick-and-place performance as long as the arm motion is sufficiently reliable. For instance, opening the drawer requires grasping the drawer and then moving the arm away from the drawer. The entire task does not rely on the hand's abilities.

Meanwhile, our second task set involving picking and reorienting six objects in hand relies only on the hand's dexterity and capabilities. When formulating the second set of tasks, we considered tasks like the door opening task from related work [28]. However, we realized that grasping a door handle is trivial and the rest of the task required stable arm motion while continuing to grasp the handle. Furthermore, the grasp did not even require the thumb (and would be possible without multi-fingered hands). This insight showed us that simulating many hand designs requires tasks that focus on hand dexterity. Picking up and reorienting objects in hand tests the hand's dexterity and choosing a variety of objects tests different types of grasps. The ring, barbell, sphere, cross3D, pen, and board required significantly different strategies from each other. Furthermore, testing these objects at different scales in simulation tested different strategies even on the same object.

Since our second set of tasks involving picking up and reorienting objects in-hand focused

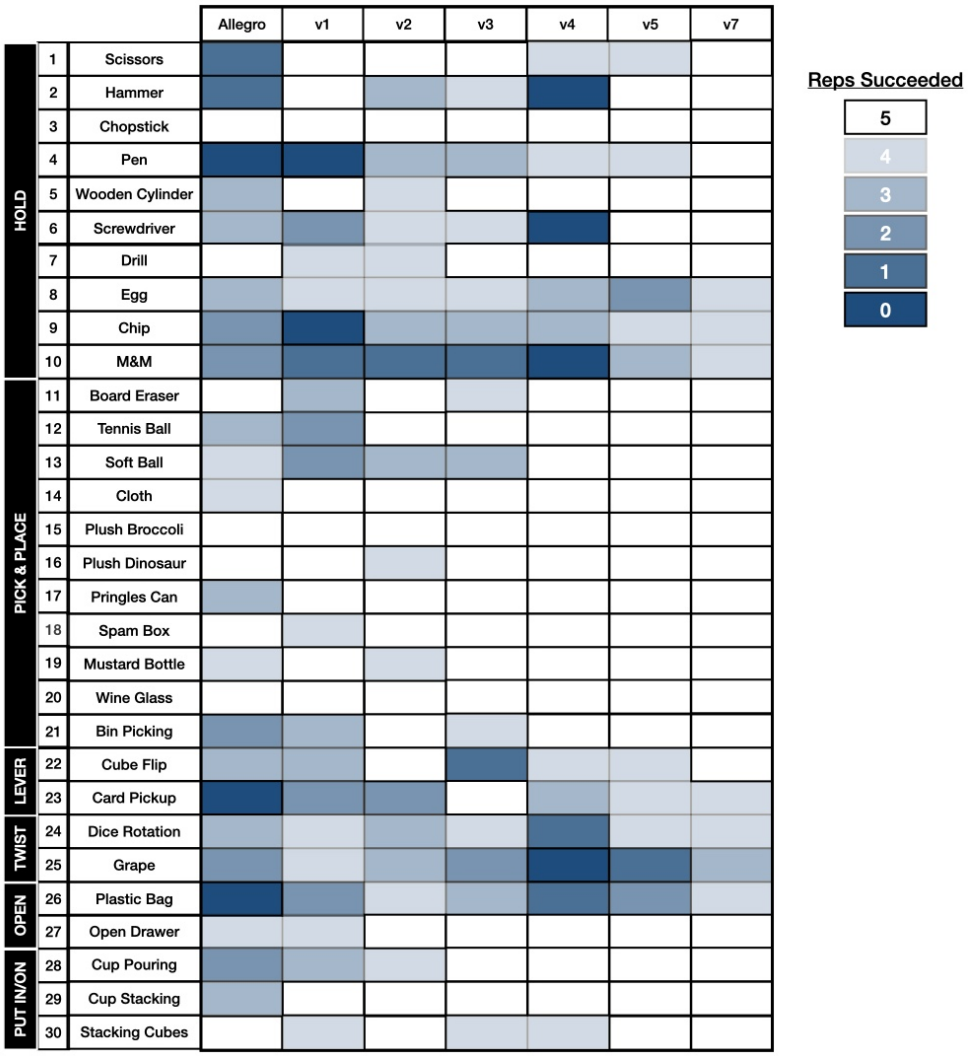



Figure 9.2: Hand performance results on DASH-30 suite of tasks for Allegro baseline hand, v1, v2, v3, v4, v5, and v7 across manipulation categories hold, pick and place, lever, twist, open, and put in/on. Each task result is shown as the number of repetitions succeeded (legend in top right corner) out of five total repetitions of each task for each hand.

specifically on dexterity and is a superset of the picking up tasks in DASH-30, we find that V1 through V5 were tested on an easier set of tasks in real-world design iteration. Then, we optimized hand designs automatically in simulation on the second harder task set and found optimized hand designs V6 and V7. Thus, in Chapter 8, we found that our best-ranking hand design in simulation V7 outperforms our manually designed hands V1 through V5. This is due to the way we chose tasks and evolved our task set to create a more difficult task set for design automation. We learned that creating tasks with a spectrum of difficulty in DASH-30 was useful for differentiating hand performance and even further design iteration was evaluated on a more difficult task set focused on dexterity to help guide a new successful hand design iteration. In order to improve the hand's capabilities beyond our existing two task suites, we could consider a new task set involving tool manipulation that would focus on the hand's ability to pick up and use objects like Hammer, Screwdriver, Drill, etc, and would most likely require stronger motors for DASH.

**Non-binary evaluation metrics that include qualitative data increases differentiation among hand design results.**

In addition to the type of tasks we evaluated, the evaluation metric was important for differentiating hand performance. For DASH-30, we had binary values of success and failure such as dropping the object instead of picking up and holding the object firmly. Meanwhile, the pickup and reorient tasks were graded on a scale of 0 to 1 based on the goal poses the objects were evaluated on. The latter provided greater granularity which we learned from the qualitative data provided by the observations from our DASH-30 evaluations. While the quantitative data we collected from DASH-30 experiments included task success per repetition of each task, aggregate data and observations helped inform successive design iterations. The types of trends and observations that were most helpful for design iteration in the real world included performance across repetitions of the same task to showcase reliability, trends in tasks the hands failed on to highlight areas for improvement, types of grasps possible or used to understand differences in grasping strategies used, and opposability and reachability of fingertips to identify limitations during task execution. We focused on grasp stability when creating evaluation metrics for picking up and reorienting objects in hand. This task suite included 6 goal poses per object where 5 of the goal poses were scored with 0, 0.25, 0.5, 0.75, or 1 where 0 is a failed grasp and 1 is a very stable grasp that is not movable with external disturbance. The last goal pose which requires setting the object down on the table is scored with 0, 0.5, or 1 where 0 is when the object is dropped onto the table from a large height and 1 is a graceful release of the object with contact with the table. Having granularity in evaluation metrics enabled differences in grasp qualities and therefore hand designs.

### 9.1.3 Rapid Prototyping and Teleoperation Evaluation

Rapid prototyping and teleoperation evaluation together enabled us to quickly iterate on hand designs in the real world, as well as verify simulated hand designs in the real world. Rapid prototyping enabled quick fabrication while teleoperation evaluated hand designs in the real-world for instant feedback to the designer. Both of these methods allowed for design iteration in

the real world, allowing us to design, evaluate, and re-design hands in a matter of days.

**Rapid prototyping can enable fast design iteration in the real world.**

Fabricating dexterous hands using rapid prototyping techniques is a well-explored field, as discussed in Chapter 2. However, quickly iterating on the design of dexterous hands in the real world is difficult without a way to rapidly prototype hands reliably. DASH was designed in a specific way that enabled fast assessment and fabrication for design iteration in the real world. Firstly, we made DASH as a soft tendon-driven multi-fingered hand. This led to a design with all of the motors below the palm, removing any need for motors at each of the joints due to tendon-driven actuation, and modularity. All of the fingers are identical and both soft and rigid parts were 3D-printed. This allowed for less human intervention and assembly, and enabled parts to be swapped out as needed. For example, V4 and V5 were identical other than finger geometry so the fingers can be replaced to change between the two designs. Similarly, V6 and V7 share similar CAD models except for the palm and top plate. This way the fingers and remaining parts can be reused and reassembled to switch between the hand designs. While this added extra convenience, we found that printing time was the biggest delay in our designing, printing, and assembling pipeline for manufacturing. Parallelization in print jobs allowed us to create V2 and V3 in 3 and 5 days respectively, including teleoperation evaluation, redesigning, and manufacturing time. Rapid prototyping significantly sped up manufacturing time to enable our fast design iteration process in the real world.

**Teleoperation evaluation can give designers instant feedback about a hand design's limitations.**

We use teleoperation as our evaluation method in many of our experiments including Chapters 4, 5, 7, and 8. Teleoperation allows for quick assessment of a manipulator's capabilities that does not require hand-picking trajectories or key-framed poses. Design limitations due to hand kinematics and compliance are very apparent with teleoperating the fabricated hand. The poor fingertip reachability of V1 and high finger stiffness (of MCP joints) of V4 were immediate observations from the teleoperator even before performing the benchmark manipulation experiments. Similarly, the teleoperator is the designer for the robotic hands, which adds the designer into the design iteration feedback loop. The designer is able to collect observations and insights into each design that can inform future design iterations. Additionally, teleoperation allows us to use similar control for new hand designs rather than learning new control policies similar to what we did for simulation.

In our experiments, we found that teleoperation was able to perform complex manipulation tasks with our hand designs, possibly even better than autonomous policies. One of the main insights is that human intuition plays a large role in being able to adjust grasps and try many grasping strategies that would be difficult for autonomous policy learning. This points to a new exciting direction where teleoperated demonstrations could help us learn better autonomous control policies for manipulation. However, this technique may not have been possible before newly developed VR technologies like Manus, which can achieve $0.1$-degree accuracy to allow for complex teleoperated manipulations.

**Teleoperation limitations include human bias and learning effects.**

Relying on human operators can result in certain biases and limitations in terms of manipulation task performance. We rely on the teleoperator's ability to compensate for the robotic hand designs flaws. For example, if picking up an object with the index and thumb fingers is not possible, the teleoperator is allowed to complete the task using a cigarette grasp (between the index and middle finger). Any way to solve the task is acceptable. We are mainly concerned about the hand design's ability to complete the task at all, where observations or experiment results will show the trends of failed tasks or types of grasps possible. The teleoperator could find a single way to solve the task and repeat it exactly the same way across repetitions or try different strategies across repetitions. This is one limitation of teleoperation evaluation. Furthermore, the designer in the loop will be geared towards human-like features or designs due to the hand being controlled by their own hand. It is natural to aim for the human hand as the ideal design for easier teleoperation.

As with any skill, practice makes perfect. While we allowed teleoperators to take one hour for practice to adjust to a new hand design, this time shortens across iterations as knowledge is transferrable to similar designs (due to learning bias). When designs diverge too much from human hands, the learning time will increase. The DASH-30 task suite mainly consisted of pick and place tasks which did not show large learning bias when repeating V3 experiments after evaluating all five design iterations V1 to V5 (see Chapter 8). However, with our second task suite of picking up and reorienting objects in-hand, we observed that the teleoperators required more practice and expertise to perform these tasks. To avoid the learning bias as the teleoperator gains experience across hand iteration experiments, we repeat the hands in the reverse order and average results. We found that our pick up and reorient tasks performed in the reverse direction did indeed perform better. Taking the average and maximum of these scores was useful to understand the peak performance of these hands as well as their relative performance to other hand designs. In addition, our preliminary results with two other users performing the pick up and reorient task showed that the average and maximum scores corresponded to our intuitions. The more human-like hand had the best average score due to ease of teleoperation and the optimized hand had the best maximum score due to enhanced capabilities. We note that new operators are ideal teleoperators if the goal is to find the easiest hand design to operate by non-experts. However, in our case, having the same expert teleoperator perform all of our manipulation experiments was key to fully evaluating the hand designs capabilities.

### 9.1.4 Simulated Hand Designs

Simulating hand designs can explore a much larger design space than manual design iteration. We were able to test only five design iterations manually but hundreds in simulation. Hand design parameter ranges were still influenced from designer intuition from manual design iteration which was a local design space search that provided intuition for our larger automated design iteration. In our work, simulating hands expedited design iteration by testing many hands, some of which we verify in the real world.

**Policy transfer and genetic algorithms can enable learning dexterous skills for many new hand designs.**

The main barrier to testing many designs in simulation is learning dexterous skills for each new hand design out of many designs in a large parameter search space. We found that policy transfer and genetic algorithms can be used to both learn policies for hand designs and generate new hand designs from existing ones. Robot interpolation enables us to sample intermediate hand design candidates between a source and target robot. We begin by learning the expert policy for the manipulation task suite on a ShadowHand-like structure (without the pinky). Then, we transfer the expert policy on that source hand to a target hand like V1. Intermediate hand designs are morphed versions of the source hand on the evolution to become the target hand. These are also considered valid robot hand designs. This allows us to incrementally transfer control policies to nearby designs and eventually to the target robot hand. Genetic algorithms enable crossovers and mutations of hand designs to generate new designs on top of existing candidates such as V1 to V5. Transferring policies is an efficient way to learn new policies for our new hand designs, especially since we are searching hand designs that are relatively close to each other (they all have four fingers and a palm).

**Simulated hand design results can resemble real world hand design performance in terms of relative ranking.**

While we simulated our hand designs as rigid bodies and fabricated them as soft robotic hands, we were curious whether our simulated hand performance would resemble real world hand performance. We found that the general ranking of hand performance was similar, such as V6 performing better than V7 on most of the pick up and reorient tasks. The optimized hands, V6 and V7, performed similarly with respect to each other and the manually iterated hand designs V1 through V5. We restricted the hand design parameter search to be feasible designs we could produce, limited torque limits, and added friction tape to real world objects. Our simulated object interactions assumed point contacts but the compliance of our soft hands enabled contact-rich manipulations. Simulating hand designs resulted in our optimized hand design V7 which outperformed our manually iterated hand design V5 by mastering four more tasks (see Figure 8.1). This jump in capabilities did not happen across V2 to V5. V1 to V2 performance was similar in adding four new capabilities but the manual design iteration started to plateau in improvement. Simulating hand designs expedited our design iteration process.

While we tested simulated hand designs using teleoperation on the same tasks, we could use teleoperation demonstrations of tasks in order to learn expert policies (instead of learning from scratch). This could also help close the gap between sim and real when transferring policies from simulation to the real robot.

**Testing simulated hand designs in the real world is useful to close the gap and understand simulation limitations.**

The sim to real gap from automating design iteration required real world evaluation to determine whether the optimized designs' performance in simulation transferred to real-world hand performance. We observed that features like fingertip shape did not resemble real-world hand

performance. We also added elements like friction tape and foam to our objects for real-world manipulation experiments even though we 3D-printed the six objects from the same simulation specifications. Despite gaps between simulated and real world hand performance, such as simulated rigid bodies and fabricated soft hands, we found similarities in our teleoperated manipulation experiments and simulated hand design performance. Related work shows that FEA is tedious and slow but can be replaced by simulating key-framed poses to inform design choices [7]. We take this a step further by showing that simulating learned manipulation policies can inform design choices in the real world. And teleoperation evaluation allowed us to assess these designs in the real world without transferring the exact policies to the real world and relying on human operators to learn the best policy for each hand for each object tested.

# Chapter 10

# Conclusion and Future Work

## 10.1 Conclusions

This thesis explores techniques for iterative design including rapid prototyping, teleoperation, and simulation to enable designers to tune kinematics and compliance for dexterous tasks through real robot evaluation. We show that substantial improvements can be made between design iterations and over state of the art dexterous robotic hands on dexterity benchmarks. Our insights about design parameters reveal that hand kinematics and compliance alone can improve dexterity rather than adding motors or fingers. We found that a variety of tasks and objects as well as non-binary evaluation metrics help difference hand design performance. All of our real world evaluation and iteration was possible through rapid prototyping and quick teleoperation assessment of the hand designs' capabilities. Lastly, we efficiently learn new dexterous skills in simulation through policy transfer and find ranked hand performance in simulation correlates with real world performance.

## 10.2 Future Work

In this section, we cover future extensions of our work in the following areas: design parameters, tasks, teleoperation evaluation, rapid prototyping, and simulation.

### 10.2.1 Design Parameters

The design parameter features we tested in simulation include palm size, finger placement (position and orientation), and finger length. In addition to these features, manual design iteration also included finger geometry such as joint stiffness, crease width, and fingertip shape. Finger geometry is a high-dimensional parameter space that was not thoroughly explored in this work and had significant effects on task performance even between thin wedge-like versus rounded spherical fingertips. While our automated design iteration tested hundreds of designs in simulation, we did not study the relationship between the parameter space and hand performance. For instance, understanding whether designs between V1 and V2 perform within the range of the two hands' scores or sensitivity experiments where changing the designs by small amounts results

in similar or different results. This could help us understand if our evaluation metric is good and whether there is an inductive bias. Generally, we found that thumb placement is the most sensitive, followed by fingertip shape, and that length of finger was least sensitive (within some range). We leave it to future work to corroborate this in simulation.

### 10.2.2 Tasks

Experiments in this thesis were limited to our manipulation task suites such as DASH-30 and in-hand reorientation of specific objects. While our curated tasks were useful to differentiate our hand designs for iteration, further work can be done to find a minimal set of tasks that provides good signal for design iteration without drastically increasing evaluation time. Additionally, each category of tasks can be balanced with the same number of tasks to demonstrate a well-rounded picture of the hand's capabilities. The axes we explored include object size, weight, and material as well as categories of task such as pick-and-place, hold, lever, etc. Within a minimal set of tasks, we can cover more types of tasks and objects efficiently to better cover grasp taxonomies [19] or everyday taxonomies [21] to further evaluate the capabilities of our hands.

### 10.2.3 Rapid Prototyping

The designs of our manipulators presented in this thesis are prototypes that can redesigned for easier fabrication and use. Specifically, DASH uses tendon-driven actuation which can be difficult to replace without disassembling at least parts of the hand. Improving the modularity and replaceability of tendons would greatly improve the user experience of DASH. The tendon routing is an S-shape through the top plate but would be easier to assemble with straight lines. Replacing tendons without disassembling would be possible by improving the tendon routing in the top plate. Furthermore, the amount of payload and finger pullout strength is strictly dependent on the motors we use. We did not change the motors at all during design iteration but increasing the motor stall torque would improve the payload of the hand and could even enable forceful manipulations such as tool-based tasks. Our main bottleneck for fabricating DASH hands was printing time, reducing this time or parallelizing prints efficiently is left to future work.

### 10.2.4 Teleoperation Evaluation

Teleoperation evaluation has some open challenges, such as overcoming human bias and learning effects. Firstly, human bias can lead teleoperators to perform poorly on hands that are not as human-like or intuitive to control which can artificially lower scores on potentially good hand designs. In this work, we use joint-to-joint mapping between the teleoperator's glove and the robot hand, but using pose-to-pose mapping could alleviate these issues by allowing the teleoperator to pose in the same manner that is expected of the teleoperated robot hand. Furthermore, teleoperators are allowed to try different grasping strategies across repetitions of the same task. This can add human error in terms of trying a bad strategy instead of repeating a good known strategy. However, this allowed us to have variety in our evaluation scores rather than always

failing or always succeeding on a task across repetitions. Secondly, learning effects disproportionately affected the in-hand reorientation tasks rather than the DASH-30 tasks which mostly involved grasping. Learning effects are most significant for difficult tasks that require lots of practice (such as in-hand reorientation). Future work on mitigating human bias and learning effects can include replaying teleoperated trajectories to evaluate task robustness. This can leverage human intuition and exploration to find the best policy for a given hand and task but removes human intervention. Eventually, teleoperation demonstrations or trajectories can inform learned policies which is our end goal for autonomously controlling dexterous hands. Additionally, our teleoperation evaluations were open-loop in that teleoperators only had visual feedback to operate the robotic hand. Future work incorporating sensing in the hands will enable closed-loop manipulation for tasks like M&M and Chip which were difficult for our optimized hand designs but could bridge the gap between achieving 23 versus all of the DASH-30 tasks.

### 10.2.5 Simulation

The sim to real gap is incredibly challenging and we only explored a small part of it. Our simulated hand designs interacted with smooth rigid objects in simulation and the real world. However, our real world 3D-printed objects from the same simulated object specifications were too difficult to manipulation as is. We used friction tape and foam in order to increase the ability to grasp and reorient the objects in-hand. In the future, we would like to have contact-rich manipulations in simulation to better bridge the gap between sim and real environments. Especially with soft robotic hands, we were able to make contact with the table and perform difficult tasks such as picking up a pen from the table in the real world. Furthermore, design features such as fingertip shape did not accurately correlate with real world results which could be due to the fast that object interactions were not realistic. Contact information can also improve optimization criteria such as evaluating power grasps and precision grasps in simulation as well as the real world (we only evaluated these grasps in the real world). Incorporating contact information into our automated design iteration method would allow for more complex tasks such as tool manipulation as well.

Furthermore, we approached the design morphology and control problems individually in our co-design optimization approach. In the future, we would like to integrate both of them more closely to efficiently optimize towards new designs.

# Bibliography


[1] Franka emika. `https://www.franka.de`. Accessed: 2023-03-27. (document), 1.1

[2] Dong-Hyuk Lee, Jae-Han Park, Sung-Woo Park, Moon-Hong Baeg, and Ji-Hun Bae. Kitech-hand: A highly dexterous and modularized robotic hand. *IEEE/ASME Transactions on Mechatronics*, 22(2):876–887, 2016. (document), 1.1, 2.1, 2.4, 5.1, 5.5

[3] Shadow dexterous hand price list. https://www.shadowrobot.com/dexterous-hand-series/. Accessed: 2021-02-28. (document), 1.1, 2.1, 4.1

[4] Alexander Buryanov and Viktor Kotiuk. Proportions of hand segments. *Int. J. Morphol*, pages 755–758, 2010. (document), 9.1.1, 9.1

[5] Healthline. https://www.healthline.com/health/average-hand-size. Accessed on 2022-11-29. (document), 5.4.2, 9.1.1, 9.1

[6] Hayley McClintock, Fatma Zeynep Temel, Neel Doshi, Je-sung Koh, and Robert J Wood. The millidelta: A high-bandwidth, high-precision, millimeter-scale delta robot. *Science Robotics*, 3(14):eaar3018, 2018. 1.2, 2.4, 3.2, 4.2

[7] Dominik Bauer, Cornelia Bauer, Arjun Lakshmipathy, Roberto Shu, and Nancy S Pollard. Towards very low-cost iterative prototyping for fully printable dexterous soft robotic hands. In *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, pages 490–497. IEEE, 2022. 1.3, 2.3, 2.4, 5.1, 5.2, 5.6, 7.3, 9.1.4

[8] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *The International Journal of Robotics Research*, 41(7):690–705, 2022. 2.1

[9] Nikhil Chavan Dafle, Alberto Rodriguez, Robert Paolini, Bowei Tang, Siddhartha S Srinivasa, Michael Erdmann, Matthew T Mason, Ivan Lundberg, Harald Staab, and Thomas Fuhlbrigge. Extrinsic dexterity: In-hand manipulation with external forces. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1578–1585. IEEE, 2014. 2.1

[10] Wenxuan Zhou and David Held. Learning to grasp the ungraspable with emergent extrinsic dexterity. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 150–160. PMLR, 14–18 Dec 2023. 2.1

[11] C Piazza, G Grioli, MG Catalano, and AJAROC Bicchi. A century of robotic hands.

*Annual Review of Control, Robotics, and Autonomous Systems*, 2, 2019. 2.1, 7.1

[12] Raunaq Bhirangi, Abigail DeFranco, Jacob Adkins, Carmel Majidi, Abhinav Gupta, Tess Hellebrekers, and Vikash Kumar. All the feels: A dexterous hand with large area sensing. *arXiv preprint arXiv:2210.15658*, 2022. 2.1

[13] Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *arXiv preprint arXiv:2309.06440*, 2023. 2.1, 2.4

[14] Michael Ahn, Henry Zhu, Kristian Hartikainen, Hugo Ponte, Abhishek Gupta, Sergey Levine, and Vikash Kumar. Robel: Robotics benchmarks for learning with low-cost robots. In *Conference on robot learning*, pages 1300–1313. PMLR, 2020. 2.1

[15] Dominik Bauer, Cornelia Bauer, Jonathan P King, Daniele Moro, Kai-Hung Chang, Stelian Coros, and Nancy Pollard. Design and control of foam hands for dexterous manipulation. *International Journal of Humanoid Robotics*, 17(01):1950033, 2020. 2.1, 2.3, 2.4

[16] Steffen Puhlmann, Jason Harris, and Oliver Brock. Rbo hand 3: A platform for soft dexterous manipulation. *IEEE Transactions on Robotics*, 38(6):3434–3449, 2022. 2.1, 2.4

[17] Cristina Piazza, Manuel G Catalano, Sasha B Godfrey, Matteo Rossi, Giorgio Grioli, Matteo Bianchi, Kristin Zhao, and Antonio Bicchi. The softhand pro-h: a hybrid body-controlled, electrically powered hand prosthesis for daily living and working. *IEEE Robotics & Automation Magazine*, 24(4):87–101, 2017. 2.1

[18] Cecilia Laschi, Barbara Mazzolai, and Matteo Cianchetti. Soft robotics: Technologies and systems pushing the boundaries of robot abilities. *Science robotics*, 1(1):eaah3690, 2016. 2.1, 3.1, 4.1

[19] Thomas Feix, Javier Romero, Heinz-Bodo Schmiedmayer, Aaron M Dollar, and Danica Kragic. The grasp taxonomy of human grasp types. *IEEE Transactions on human-machine systems*, 46(1):66–77, 2015. 2.2, 7.1, 10.2.2

[20] Adalbert I Kapandji. Clinical evaluation of the thumb's opposition. *Journal of Hand Therapy*, 5(2):102–106, 1992. 2.2

[21] Jia Liu, Fangxiaoyu Feng, Yuzuko C. Nakamura, and Nancy S. Pollard. A taxonomy of everyday grasps in action. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 573–580, 2014. doi: 10.1109/HUMANOIDS.2014.7041420. 2.2, 5.3.4, 10.2.2

[22] Ryan Coulson, Chao Li, Carmel Majidi, and Nancy S Pollard. The elliott and connolly benchmark: A test for evaluating the in-hand dexterity of robot hands. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 238–245. IEEE, 2021. 2.2

[23] Johanne Desrosiers, Réjean Hébert, Gina Bravo, and Elisabeth Dutil. The purdue pegboard test: normative data for people aged 60 and over. *Disability and rehabilitation*, 17 (5):217–224, 1995. 2.2

[24] Virgil Mathiowetz, Gloria Volland, Nancy Kashman, and Karen Weber. Adult norms for the box and block test of manual dexterity. *The American journal of occupational therapy*, 39(6):386–391, 1985. 2.2

[25] Erika Davis Sears and Kevin C Chung. Validity and responsiveness of the jebsen–taylor hand function test. *The Journal of hand surgery*, 35(1):30–37, 2010. 2.2

[26] Jo Adams, Kate Hodges, Joanna Kujawa, and Cheryl Metcalf. Test-retest reliability of the southampton hand assessment procedure. *International Journal of Rehabilitation Research*, 32:S18, 2009. 2.2

[27] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha S Srinivasa, Pieter Abeel, and Aaron M Dollar. Ycb benchmarking project: Object set, data set and their applications. *Journal of The Society of Instrument and Control Engineers*, 56 (10):792–797, 2017. 2.2, 7.1

[28] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017. 2.2, 7.1, 7.5.1, 9.1.2

[29] Boling Yang, Patrick E. Lancaster, Siddhartha S. Srinivasa, and Joshua R. Smith. Benchmarking robot manipulation with the rubik's cube. *IEEE Robotics and Automation Letters*, 5(2):2094–2099, 2020. doi: 10.1109/LRA.2020.2969912. 2.2, 7.1

[30] Silvia Cruciani, Balakumar Sundaralingam, Kaiyu Hang, Vikash Kumar, Tucker Hermans, and Danica Kragic. Benchmarking in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(2):588–595, 2020. 2.2

[31] Joseph Falco, Karl Van Wyk, and Elena Messina. Performance metrics and test methods for robotic hands. *DRAFT NIST Special Publication*, 1227, 2018. 2.2

[32] Julian Whitman, Matthew Travers, and Howie Choset. Modular mobile robot design selection with deep reinforcement learning. In *NeurIPS Workshop on ML for engineering modeling, simulation and design*, 2020. 2.3, 7.2

[33] Charles Schaff, Audrey Sedal, and Matthew R Walter. Soft robots learn to crawl: Jointly optimizing design and control with sim-to-real transfer. *arXiv preprint arXiv:2202.04575*, 2022. 2.3

[34] Melissa Mozian, Juan Camilo Gamboa Higuera, David Meger, and Gregory Dudek. Learning domain randomization distributions for training robust locomotion policies. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6112–6117. IEEE, 2020. 2.3

[35] Steffen Puhlmann, Jason Harris, and Oliver Brock. Rbo hand 3: A platform for soft dexterous manipulation. *IEEE Transactions on Robotics*, 38(6):3434–3449, dec 2022. 2.3, 5.1, 5.2, 5.6

[36] Yong-Jae Kim, Younbaek Lee, Jiyoung Kim, Ja-Woo Lee, Kang-Min Park, Kyung-Sik Roh, and Jung-Yun Choi. Roboray hand: A highly backdrivable robotic hand with sensorless contact force measurements. In *2014 IEEE International Conference on Robotics*

*and Automation (ICRA)*, pages 6712–6718. IEEE, 2014. 2.3

[37] John Amend and Hod Lipson. The jamhand: dexterous manipulation with minimal actuation. *Soft robotics*, 4(1):70–80, 2017. 2.3

[38] Walter G Bircher, Andrew S Morgan, and Aaron M Dollar. Complex manipulation with a simple robotic hand through contact breaking and caging. *Science Robotics*, 6(54): eabd2666, 2021. 2.3

[39] Christopher Hazard, Nancy Pollard, and Stelian Coros. Automated design of robotic hands for in-hand manipulation tasks. *International Journal of Humanoid Robotics*, 17(01): 1950029, 2020. 2.3

[40] Zhong Zhang, Yu Zheng, Zhe Hu, Lezhang Liu, Xuan Zhao, Xiong Li, and Jia Pan. A computational framework for robot hand design via reinforcement learning. In *2021 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 7216–7222. IEEE, 2021. 2.3

[41] Tianjian Chen, Zhanpeng He, and Matei Ciocarlie. Hardware as policy: Mechanical and computational co-optimization using deep reinforcement learning. *arXiv preprint arXiv:2008.04460*, 2020. 2.3

[42] Jie Xu, Tao Chen, Lara Zlokapa, Michael Foshey, Wojciech Matusik, Shinjiro Sueda, and Pulkit Agrawal. An end-to-end differentiable framework for contact-aware robot design. *arXiv preprint arXiv:2107.07501*, 2021. 2.3

[43] Raphael Deimel, Patrick Irmisch, Vincent Wall, and Oliver Brock. Automated co-design of soft hand morphology and control strategy for grasping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1213–1218. IEEE, 2017. 2.3, 5.2, 7.2

[44] Andre Meixner, Christopher Hazard, and Nancy Pollard. Automated design of simple and robust manipulators for dexterous in-hand manipulation tasks using evolutionary strategies. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pages 281–288. IEEE, 2019. 2.3, 7.2

[45] Raphael Deimel and Oliver Brock. A compliant hand based on a novel pneumatic actuator. In *2013 IEEE International Conference on Robotics and Automation*, pages 2047–2053. IEEE, 2013. 2.4

[46] Raymond R Ma, Lael U Odhner, and Aaron M Dollar. A modular, open-source 3d printed underactuated hand. In *2013 IEEE International Conference on Robotics and Automation*, pages 2737–2743. IEEE, 2013. 2.4

[47] Jahan Zeb Gul, Memoon Sajid, Muhammad Muqeet Rehman, Ghayas Uddin Siddiqui, Imran Shah, Kyung-Hwan Kim, Jae-Wook Lee, and Kyung Hyun Choi. 3d printing for soft robotics–a review. *Science and technology of advanced materials*, 19(1):243–262, 2018. 2.4

[48] Raphael Deimel and Oliver Brock. A novel type of compliant and underactuated robotic hand for dexterous grasping. *The International Journal of Robotics Research*, 35(1-3): 161–185, 2016. doi: 10.1177/0278364915592961. URL `https://doi.org/10.`

1177/0278364915592961. 2.4

[49] Henry Zhu, Abhishek Gupta, Aravind Rajeswaran, Sergey Levine, and Vikash Kumar. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3651–3657. IEEE, 2019. 2.4, 4.1, 5.3.2

[50] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022. 2.4, 6.3.1, 6.5

[51] L Rey and R Clavel. The delta parallel robot. In *Parallel Kinematic Machines: Theoretical Aspects and Industrial Requirements*, pages 401–417. Springer, 1999. 3.1, 3.3

[52] Lung-Wen Tsai. *Robot analysis: the mechanics of serial and parallel manipulators*. John Wiley & Sons, 1999. 3.1, 3.2

[53] Nicola Pio Belfiore and Pasquale Simeone. Inverse kinetostatic analysis of compliant four-bar linkages. *Mechanism and Machine Theory*, 69:350–372, 2013. 3.2

[54] Marco Balucani, Nicola Pio Belfiore, Rocco Crescenzi, and Matteo Verotti. The development of a mems/nems-based 3 dof compliant micro robot. In *19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010)*, pages 173–179. IEEE, 2010. 3.2

[55] Yanling Tian, Bijan Shirinzadeh, Dawei Zhang, Xianping Liu, and D Chetwynd. Design and forward kinematics of the compliant micro-manipulator with lever mechanisms. *Precision Engineering*, 33(4):466–475, 2009. 3.2

[56] Umesh Bhagat, Bijan Shirinzadeh, Leon Clark, Peter Chea, Yanding Qin, Yanling Tian, and Dawei Zhang. Design and analysis of a novel flexure-based 3-dof mechanism. *Mechanism and Machine Theory*, 74:173–187, 2014. 3.2

[57] Martin L Culpepper and Gordon Anderson. Design of a low-cost nano-manipulator which utilizes a monolithic, spatial compliant mechanism. *Precision engineering*, 28(4):469–482, 2004. 3.2

[58] Byung-Ju Yi, Goo Bong Chung, Heung Yeol Na, Whee Kuk Kim, and Il Hong Suh. Design and experiment of a 3-dof parallel micromechanism utilizing flexure hinges. *IEEE Transactions on robotics and automation*, 19(4):604–612, 2003. 3.2

[59] Ankur Mehta, Joseph DelPreto, and Daniela Rus. Integrated codesign of printable robots. *Journal of Mechanisms and Robotics*, 7(2):021015, 2015. 3.2

[60] Suk-Jun Kim, Dae-Young Lee, Gwang-Pil Jung, and Kyu-Jin Cho. An origami-inspired, self-locking robotic arm that can be folded flat. *Science Robotics*, 3(16):eaar2915, 2018. 3.2

[61] ByungHyun Shin, Samuel M Felton, Michael T Tolley, and Robert J Wood. Self-assembling sensors for printable machines. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4417–4422. IEEE, 2014. 3.2

[62] Je-sung Koh, Daniel M Aukes, Brandon Araki, Sarah Pohorecky, Yash Mulgaonkar, Michael T Tolley, Vijay Kumar, Daniela Rus, and Robert J Wood. A modular folded laminate robot capable of multi modal locomotion. In *2016 International Symposium on*

*Experimental Robotics*, pages 59–70. Springer, 2017. 3.2

[63] Jorge E Correa, Joseph Toombs, Nicholas Toombs, and Placid M Ferreira. Laminated micro-machine: Design and fabrication of a flexure-based delta robot. *Journal of Manufacturing Processes*, 24:370–375, 2016. 3.2, 4.2

[64] Ryan St Pierre, Noah Paul, and Sarah Bergbreiter. 3dflex: A rapid prototyping approach for multi-material compliant mechanisms in millirobots. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3068–3073. IEEE, 2017. 3.2

[65] Doug Stewart. A platform with six degrees of freedom. *Proceedings of the institution of mechanical engineers*, 180(1):371–386, 1965. 3.2

[66] Jean-Pierre Merlet. *Parallel robots*, volume 128. Springer Science & Business Media, 2005. 4.1

[67] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. 4.1, 5.2, 6.2

[68] Josie Hughes, Utku Culha, Fabio Giardina, Fabian Guenther, Andre Rosendo, and Fumiya Iida. Soft manipulators and grippers: a review. *Frontiers in Robotics and AI*, 3:69, 2016. 4.1

[69] John R Amend, Eric Brown, Nicholas Rodenberg, Heinrich M Jaeger, and Hod Lipson. A positive pressure universal gripper based on the jamming of granular material. *IEEE transactions on robotics*, 28(2):341–350, 2012. 4.1

[70] Robotiq. `https://thinkbotsolutions.com/collections/robotiq`. Accessed: 2021-04-21. 4.1

[71] Daniela Rus and Michael T Tolley. Design, fabrication and control of soft robots. *Nature*, 521(7553):467–475, 2015. 4.2

[72] Pragna Mannam, Oliver Kroemer, and F. Zeynep Temel. Characterization of compliant parallelogram links for 3d-printed delta manipulators. *Proceedings of International Symposium on Experimental Robotics (ISER '20)*, March 2021. 4.2, 4.3.1, 4.3.2

[73] Mohamed Bouri and Reymond Clavel. The linear delta: Developments and applications. In *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, pages 1–8. VDE, 2010. 4.2

[74] Ryan L Truby, Cosimo Della Santina, and Daniela Rus. Distributed proprioception of 3d configuration in soft, sensorized robots via deep learning. *IEEE Robotics and Automation Letters*, 5(2):3299–3306, 2020. 4.2

[75] Bianca S Homberg, Robert K Katzschmann, Mehmet R Dogar, and Daniela Rus. Haptic identification of objects using a modular soft robotic gripper. pages 1698–1705, 2015. 4.2

[76] Gabor Soter, Andrew Conn, Helmut Hauser, and Jonathan Rossiter. Bodily aware soft robots: integration of proprioceptive and exteroceptive sensors. pages 2448–2453, 2018. 4.2

[77] Ultimaker: Materials. https://ultimaker.com/materials. Accessed: 2021-02-27. 4.3

[78] M López, E Castillo, G García, and A Bashir. Delta robot: Inverse, direct, and intermediate jacobians. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 220(1):103–109, 2006. doi: 10.1243/095440606X78263. URL https://doi.org/10.1243/095440606X78263. 4.4

[79] B. Mehrafrooz, M. Mohammadi, and M. T. Masouleh. Kinematic sensitivity evaluation of revolute and prismatic 3-dof delta robots. In *2017 5th RSI International Conference on Robotics and Mechatronics (ICRoM)*, pages 225–231, 2017. doi: 10.1109/ICRoM.2017.8466159. 4.4

[80] Michael I. Jordan and David E. Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16(3):307–354, 1992. ISSN 0364-0213. doi: https://doi.org/10.1016/0364-0213(92)90036-T. URL https://www.sciencedirect.com/science/article/pii/036402139290036T. 4.4.2

[81] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170. IEEE, 2020. 4.6, 6.3.1

[82] Feifei Chen and Michael Yu Wang. Design optimization of soft robots: A review of the state of the art. *IEEE Robotics & Automation Magazine*, 27(4):27–43, 2020. 5.1, 5.2

[83] Cosimo Della Santina, Cristina Piazza, Giorgio Grioli, Manuel G Catalano, and Antonio Bicchi. Toward dexterous manipulation with augmented adaptive synergies: The pisa/iit softhand 2. *IEEE Transactions on Robotics*, 34(5):1141–1156, 2018. 5.1

[84] Yu Zhang and Robert K Katzschmann. Creation of a modular soft robotic fish testing platform. *arXiv preprint arXiv:2201.04098*, 2022. 5.1, 5.1, 5.2

[85] Sylvain Abondance, Clark B Teeple, and Robert J Wood. A dexterous soft robotic hand for delicate in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(4):5502–5509, 2020. 5.1

[86] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 1749–1759, October 2021. 5.1, 6.2, 6.3.1, 6.3.1

[87] Aravind Sivakumar, Kenneth Shaw, and Deepak Pathak. Robotic telekinesis: learning a robotic hand imitator by watching humans on youtube. *arXiv preprint arXiv:2202.10448*, 2022. 5.1, 5.3.4, 6.2, 6.3.1

[88] Marco Controzzi, Christian Cipriani, and Maria Chiara Carrozza. *Design of Artificial Hands: A Review*, pages 219–246. Springer International Publishing, Cham, 2014. ISBN 978-3-319-03017-3. doi: 10.1007/978-3-319-03017-3_11. URL https://doi.org/10.1007/978-3-319-03017-3_11. 5.2

[89] Naishi Feng, Qiurong Shi, Hong Wang, Jiale Gong, Chong Liu, and Zhiguo Lu. A soft

robotic hand: design, analysis, semg control, and experiment. *The International Journal of Advanced Manufacturing Technology*, 97:319–333, 2018. 5.2

[90] Yahya Elsayed, Augusto Vincensi, Constantina Lekakou, Tao Geng, CM Saaj, Tommaso Ranzani, Matteo Cianchetti, and Arianna Menciassi. Finite element analysis and design optimization of a pneumatically actuating silicone module for robotic surgery applications. *Soft Robotics*, 1(4):255–262, 2014. 5.2

[91] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pages 1101–1112. PMLR, 2020. 5.2

[92] Sridhar Pandian Arunachalam, Sneha Silwal, Ben Evans, and Lerrel Pinto. Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation, 2022. URL `https://arxiv.org/abs/2203.13251`. 5.2, 7.2

[93] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos. In *Conference on Robot Learning (CoRL)*, 2022. 5.2, 7.2

[94] Rui Li, Hongyu Wang, and Zhenyu Liu. Survey on mapping human hand motion to robotic hands for teleoperation. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(5):2647–2665, 2022. doi: 10.1109/TCSVT.2021.3057992. 5.2

[95] Minas V Liarokapis, Panagiotis K Artemiadis, and Kostas J Kyriakopoulos. Telemanipulation with the dlr/hit ii robot hand using a dataglove and a low cost force feedback device. In *21st Mediterranean Conference on Control and Automation*, pages 431–436. IEEE, 2013. 5.2

[96] Ninjaflex edge. https://ninjatek.com/shop/edge/. Accessed on 2022-11-26. 5.3.2, 7.3

[97] Ufactory xarm6. https://www.ufactory.cc/product-page/ufactory-xarm-6, . Accessed on 2023-02-03. 5.3.2

[98] Manus. https://www.manus-meta.com, note=Accessed on 2022-11-28. 5.3.3, 5.6, 7.6

[99] Steam vr. https://www.steamvr.com/en/. Accessed on 2023-02-03. 5.3.3

[100] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Dexpilot: Vision based teleoperation of dexterous robotic hand-arm system, 2019. URL `https://arxiv.org/abs/1910.03135`. 5.3.4, 6.2

[101] Pragna Mannam, Kenneth Shaw, Dominik Bauer, Jean Oh, Deepak Pathak, and Nancy Pollard. Designing anthropomorphic soft hands through interaction. In *Humanoids 2023*, 2023. 6.1, 7.1, 7.2, 7.3, 7.3, 7.3, 7.6, 7.6, 7.6, 7.7

[102] David F Fouhey, Xiaolong Wang, and Abhinav Gupta. In defense of the direct perception of affordances. *arXiv preprint arXiv:1505.01085*, 2015. 6.1

[103] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. 2023. 6.1, 6.3.1

[104] Mohit Goyal, Sahil Modi, Rishabh Goyal, and Saurabh Gupta. Human hands as probes for interactive object understanding. In *CVPR*, 2022. 6.1, 6.3.1

[105] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction hotspots from video. In *ICCV*, 2019. 6.1, 6.3.1

[106] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9869–9878, 2020. 6.1, 6.2, 6.3.1

[107] Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *CVPR*, 2022. 6.1, 6.3.1

[108] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21013–21022, June 2022. 6.1, 6.3.1

[109] Xiaolong Wang, Rohit Girdhar, and Abhinav Gupta. Binge watching: Scaling affordance learning from sitcoms. In *CVPR*, 2017. 6.1

[110] Todor Davchev, Kevin Sebastian Luck, Michael Burke, Franziska Meier, Stefan Schaal, and Subramanian Ramamoorthy. Residual learning from demonstration. *arXiv preprint arXiv:2008.07682*, 2020. 6.1

[111] Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *ICRA*, 2019. 6.1, 6.3.2

[112] Sertac Karaman, Matthew R Walter, Alejandro Perez, Emilio Frazzoli, and Seth Teller. Anytime motion planning using the rrt. In *ICRA*, 2011. 6.2

[113] James J Kuffner and Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 995–1001. IEEE, 2000. 6.2

[114] Mustafa Mukadam, Xinyan Yan, and Byron Boots. Gaussian process motion planning. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 9–15. IEEE, 2016. 6.2

[115] Jens Kober and Jan Peters. Policy search for motor primitives in robotics. *Advances in neural information processing systems*, 21, 2008. 6.2

[116] Jan Peters, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010. 6.2

[117] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *ICLR*, 2016. 6.2

[118] Ivaylo Popov, Nicolas Heess, Timothy Lillicrap, Roland Hafner, Gabriel Barth-Maron, Matej Vecerik, Thomas Lampe, Yuval Tassa, Tom Erez, and Martin Riedmiller. Data-efficient deep reinforcement learning for dexterous manipulation. *arXiv preprint arXiv:1704.03073*, 2017. 6.2

[119] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017. 6.2

[120] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with large-scale data collection. In *ISER*, 2016. 6.2

[121] Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In *NeurIPS*, pages 9191–9200, 2018. 6.2

[122] Pulkit Agrawal, Ashvin Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. *NIPS*, 2016. 6.2

[123] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1352–1361. JMLR. org, 2017. 6.2

[124] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018. 6.2

[125] Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021. 6.2

[126] Roberto Martin-Martin, Michelle A. Lee, Rachel Gardner, Silvio Savarese, Jeannette Bohg, and Animesh Garg. Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks. *IROS*, 2019. 6.2

[127] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *NeurIPS*, 2017. 6.2

[128] Sergey Levine and Vladlen Koltun. Guided policy search. In *ICML*, 2013. 6.2

[129] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Alan: Autonomously exploring robotic agents in the real world. *arXiv preprint arXiv:2302.06604*, 2023. 6.2

[130] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), November 2017. 6.2, 6.3.1

[131] Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. CPF: Learning a contact potential field to model the hand-object interaction. In *ICCV*, 2021. 6.2

[132] Jiayi Wang, Franziska Mueller, Florian Bernard, Suzanne Sorli, Oleksandr Sotnychenko, Neng Qian, Miguel A Otaduy, Dan Casas, and Christian Theobalt. Rgb2hands: real-time tracking of 3d hand interactions from monocular rgb video. *ACM Transactions on Graphics (TOG)*, 39(6):1–16, 2020. 6.2

[133] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. *CoRR*, abs/1712.06584, 2017. URL `http://arxiv.org/abs/1712.06584`. 6.2

[134] Yufei Ye, Abhinav Gupta, and Shubham Tulsiani. What's in your hands? 3d reconstruction of generic objects in hands. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3895–3905, 2022. 6.2

[135] Cmu graphics lab motion capture database. `http://mocap.cs.cmu.edu/`. 6.2

[136] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013. 6.2

[137] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 813–822, 2019. 6.2

[138] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 6.2

[139] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015. 6.2

[140] Pradipto Das, Chenliang Xu, Richard F Doell, and Jason J Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2634–2641, 2013. 6.2

[141] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 6.2, 6.3.1

[142] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21013–21022, June 2022. 6.2

[143] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018. 6.2, 6.3.1

[144] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022. 6.2

[145] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022. 6.2

[146] Yuzhe Qin, Hao Su, and Xiaolong Wang. From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation, 2022. 6.2

[147] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos. In *CoRL*, 2022. 6.2

[148] Priyanka Mandikal and Kristen Grauman. Dexvip: Learning dexterous grasping with human hand pose priors from video. In *Conference on Robot Learning*, pages 651–661. PMLR, 2022. 6.2

[149] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017. 6.2

[150] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018. 6.2

[151] Priyanka Mandikal and Kristen Grauman. Dexvip: Learning dexterous grasping with human hand pose priors from video. In *Conference on Robot Learning (CoRL)*, 2021. 6.2

[152] Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. Visual dexterity: In-hand dexterous manipulation from depth. *arXiv preprint arXiv:2211.11744*, 2022. 6.2

[153] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on Robot Learning*, pages 403–415. PMLR, 2023. 6.2

[154] Gabriel B Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning. *arXiv preprint arXiv:2205.02824*, 2022. 6.2

[155] Tsuneo Yoshikawa. Dynamic manipulability of robot manipulators. *Transactions of the Society of Instrument and Control Engineers*, 21(9):970–975, 1985. 6.2

[156] Haruhiko Asada and J-JE Slotine. *Robot analysis and control*. John Wiley & Sons, 1991. 6.2

[157] Kevin M Lynch and Frank C Park. *Modern robotics*. Cambridge University Press, 2017. 6.2

[158] Changliu Liu and Masayoshi Tomizuka. Designing the robot behavior for safe human robot interactions. In *Trends in Control and Decision-Making for Human-Robot Collaboration Systems*, pages 241–270. Springer, 2017. 6.2

[159] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987. 6.2

[160] Zilin Si, Tianhong Catherine Yu, Katrene Morozov, James McCann, and Wenzhen Yuan. Robotsweater: Scalable, generalizable, and customizable machine-knitted tactile skins for robots. *arXiv preprint arXiv:2303.02858*, 2023. 6.2

[161] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017. 6.2

[162] Raunaq Bhirangi, Tess Hellebrekers, Carmel Majidi, and Abhinav Gupta. Reskin: versatile, replaceable, lasting tactile skins. *arXiv preprint arXiv:2111.00071*, 2021. 6.2

[163] Subramanian Sundaram, Petr Kellnhofer, Yunzhu Li, Jun-Yan Zhu, Antonio Torralba, and Wojciech Matusik. Learning the signatures of the human grasp using a scalable tactile glove. *Nature*, 569(7758), 2019. doi: 10.1038/s41586-019-1234-z. 6.2

[164] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Phillip Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. *arXiv preprint arXiv:2201.02605*, 2022. 6.3.1

[165] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. URL `https://arxiv.org/abs/2103.00020`. 6.3.1

[166] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 6.3.1, 6.5

[167] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. *arXiv preprint arXiv:2207.09450*, 2022. 6.3.2

[168] K. Sohn, X. Yan, and H. Lee. Learning Structured Output Representation using Deep Conditional Generative Models. In *NeurIPS*, 2015. 6.3.2

[169] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014. 6.3.2

[170] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014. 6.3.2

[171] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 6.3.2

[172] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL `http://arxiv.org/abs/1512.03385`. 6.5

[173] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *arXiv preprint arXiv:2106.01345*, 2021. 6.5

[174] Silvia Cruciani, Balakumar Sundaralingam, Kaiyu Hang, Vikash Kumar, Tucker Hermans,

and Danica Kragic. Benchmarking in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(2):588–595, apr 2020. doi: 10.1109/lra.2020.2964160. 7.1

[175] Andrew S. Morgan, Walter G. Bircher, Berk Calli, and Aaron M. Dollar. Learning from transferable mechanics models: Generalizable online mode detection in underactuated dexterous manipulation. In *ICRA 2019*, 2019. doi: 10.1109/ICRA.2019.8793727. 7.1

[176] Silvia Cruciani, Christian Smith, Danica Kragic, and Kaiyu Hang. Dexterous manipulation graphs. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2040–2047, 2018. 7.1

[177] Xingyu Liu, Deepak Pathak, and Kris M Kitani. Revolver: Continuous evolutionary models for robot-to-robot policy transfer. *arXiv preprint arXiv:2202.05244*, 2022. 7.1, 7.1, 7.4.1

[178] Xingyu Liu, Deepak Pathak, and Kris M Kitani. Herd: Continuous human-to-robot evolution for learning from human demonstration. *arXiv preprint arXiv:2212.04359*, 2022. 7.1, 7.1, 7.2, 7.4.1

[179] Yuval Davidor. *Genetic Algorithms and Robotics: A heuristic strategy for optimization*, volume 1. World Scientific, 1991. 7.1

[180] Theodore W Manikas, Kaveh Ashenayi, and Roger L Wainwright. Genetic algorithms for autonomous robot navigation. *IEEE Instrumentation & Measurement Magazine*, 10(6): 26–31, 2007. 7.1

[181] Stanislav Števo, Ivan Sekaj, and Martin Dekan. Optimization of robotic arm trajectory using genetic algorithm. *IFAC Proceedings Volumes*, pages 1748–1753, 2014. 7.1

[182] Tianjian Chen, Zhanpeng He, and Matei Ciocarlie. Co-designing hardware and control for robot hands. *Science Robotics*, 6(54):eabg2133, 2021. doi: 10.1126/scirobotics.abg2133. 7.2

[183] Ci Chen, Pingyu Xiang, Haojian Lu, Yue Wang, and Rong Xiong. $\mathrm{C}^2$:co-design of robots via concurrent networks coupling online and offline reinforcement learning, 2022. 7.2

[184] Sehoon Ha, Stelian Coros, Alexander Alspach, Joohyung Kim, and Katsu Yamane. Joint optimization of robot design and motion parameters using the implicit function theorem. In *Robotics: Science and systems*, volume 8, 2017. 7.2

[185] Charles Schaff, David Yunis, Ayan Chakrabarti, and Matthew R Walter. Jointly learning to construct and control agents using deep reinforcement learning. In *ICRA 2019*, pages 9798–9805. IEEE, 2019. 7.2

[186] Julian Whitman, Matthew Travers, and Howie Choset. Learning modular robot control policies. *IEEE Transactions on Robotics*, 39(5):4095–4113, 2023. doi: 10.1109/TRO.2023.3284362. 7.2

[187] Jie Xu, Tao Chen, Lara Zlokapa, Michael Foshey, Wojciech Matusik, Shinjiro Sueda, and Pulkit Agrawal. An end-to-end differentiable framework for contact-aware robot design. In *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, 2021. 7.2

[188] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv:1703.03864*, 2017. 7.2

[189] Yuanpei Chen, Chen Wang, Li Fei-Fei, and C. Karen Liu. Sequential dexterity: Chaining dexterous policies for long-horizon manipulation, 2023. 7.2

[190] Junggon Kim, Kunihiro Iwamoto, James J Kuffner, Yasuhiro Ota, and Nancy S Pollard. Physically based grasp quality evaluation under pose uncertainty. *IEEE Transactions on Robotics*, 29(6):1424–1439, 2013. 7.2, 7.6

[191] Manuel Bonilla, Edoardo Farnioli, Cristina Piazza, M Catalano, Giorgio Grioli, Manolo Garabini, Marco Gabiccini, and Antonio Bicchi. Grasping with soft hands. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 581–587, 2014. 7.3

[192] Robotis: Dynamixel-x. https://www.robotis.us/dynamixel-xc330-m288-t/. Accessed on 2022-11-26. 7.3

[193] Ufactory xarm7. https://www.ufactory.cc/product-page/ufactory-xarm-7, . Accessed on 2023-10-08. 7.3

[194] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021. 7.5.3