

***Human-Inspired Dexterous Manipulation
of Deformable Objects:
Towards Economically Sustainable Robotic
Textile Recycling***

David Ryan Hinwood

A Thesis Submitted for the Degree of Doctor of Philosophy of the University of Canberra

January 22, 2024

Faculty of Science and Technology



**UNIVERSITY OF
CANBERRA**

Abstract

Textile manufacturing and disposal are among the most environmentally damaging product life-cycles in the early 21st century. For example, clothing disposal in landfills can cause significant damage and is expanding due to fast-fashion business practices. In addition, as clothing decomposes it emits the greenhouse gas methane, and toxic chemicals, including dye, can leak into the soil. Recycling discarded clothing by chemically breaking it down to produce new materials, such as lyocell, has been suggested to combat some of these issues. However, it is rare to see these technologies applied on a large scale, partly due to the extensive labour required to sort discarded clothing to the appropriate recycling process. This thesis examines the utilisation of robots to mitigate labour demands, focusing on pick-and-place applications involving deformable object manipulation.

Robotic manipulation of textiles remains challenging as fabric displays a unique physical behaviour due to clothing's anisotropic nature and non-linear mechanical response. Furthermore, garments can exhibit various colours, shapes, forms and textures, making physical manipulation and visual interpretation a series of complex and multifaceted tasks. This thesis addresses dexterous manipulation, recognising that clothing can present itself in states requiring skilful manipulation, and handling garments can require intrinsic dexterous skills. As a result, several academic projects have developed end-effectors to manipulate fabric. However, these solutions are not generalised and target specific manipulation pipelines such as folding a shirt or grasping flattened material. While previous research uses heuristic human observations to inspire robotic gripper designs, these approaches do not extensively explore human behaviour and morphology in a robotic context.

This observation is what inspires the research described in this thesis. First, an investigation using anthropomorphic taxonomies defines the range of skills necessary for a generalised fabric pick-and-place solution. This device is then modelled using classical mechanics and fabricated. Then, data-driven approaches in reinforcement learning are applied to build behaviours that enable the end-effector to robustly execute environmentally constrained behaviour. Finally, this thesis provides a discussion of the conducted research and an overview of future applications.

Form B

Certificate of Authorship of Thesis

Except where clearly acknowledged in footnotes, quotations and the bibliography, I certify that I am the sole author of the thesis submitted today entitled –

Human-Inspired Dexterous Manipulation of Deformable
Objects: Towards Economically Sustainable Robotic Textile
Recycling

(Thesis title)

I further certify that to the best of my knowledge the thesis contains no material previously published or written by another person except where due reference is made in the text of the thesis.

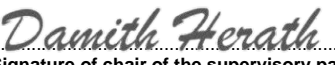
The material in the thesis has not been the basis of an award of any other degree or diploma except where due reference is made in the text of the thesis.

The thesis complies with University requirements for a thesis as set out in *Gold Book Part 7: Examination of Higher Degree by Research Theses Policy, Schedule Two (S2)*.

Refer to <http://www.canberra.edu.au/research-students/goldbook>

.....

Signature of Candidate

.....

Signature of chair of the supervisory panel

Date: 19/01/24

NOTE: The wording contained in Form B must be bound into the thesis, preferably as the third page, and signed by the author of the thesis and the supervisor. The exact layout does not need to be duplicated.

Acknowledgements

My PhD has been a long journey that would not have been possible without the support of colleagues, friends and family. It can be challenging to know who to thank first. Still, I believe the first thanks go to my wonderful parents, who let me stay with them rent-free during the first component of my research journey. Their unconditional love and support will always be appreciated while establishing research goals and life plans. I hope this research can make an impact that will make them proud. Next, the colleagues who had the most significant impact on my research journey were my supervisors, Dr Damith Herath and Prof Roland Göcke. I first met Prof Göcke as an undergraduate when he was demonstrating a NAO robot and looking for volunteers to investigate robot soccer. Having been interested in robotics since childhood, this was an opportunity I leapt at enthusiastically. Thanks to this encounter with Professor Göcke, I began my robotics journey earnestly.

I completed my Honours in Information Technology under Prof Göcke's supervision studying computer vision methods to apply within robot soccer. After finishing this course, I left with a passion for pursuing robotics further. In 2017, I then met my primary supervisor Dr Herath. So many things could be said about Dr Herath, but his resilience always stands out. Regardless of the barriers or how complex specific tasks seem, Dr Herath always knew how to break a problem down and work through it. He was also a person I spoke to regularly about creating a robotics startup, which still remains a long-term goal of mine. I cannot stress the positive impact Dr Herath's advice has given my research career. Furthermore, during early funding issues throughout my PhD, I can't think of another set of supervisors who would show the same level of support and compassion that I received. I have made it this far through the PhD experience thanks to both.

Throughout my PhD journey, I have been fortunate to work with excellent collaborators to create unique robotic social interactive exhibits internationally and locally. My first paper was with lab colleague James Ireland, who had undertaken a PhD at a similar time to myself. James has been a constant presence at the HCT Lab at UC. In addition, he has always been open to bouncing research ideas around, which I sincerely appreciate. I would also like to acknowledge Professor Elizabeth Jochum and her students Bollette, Carina, Jonas, Kristoph, Andreas and Juliana from the University of Aalborg, Denmark. Collaborating with you on a range of robotic art exhibits deployed in Australia and Denmark was an

absolute delight resulting in impressive academic publications and recognition. In addition, I want to acknowledge others, including the performance artist Stelarc, the various undergraduates I have worked with throughout my degree and the faculty IT manager Jason Weber. Finally, I would also like to thank Hayley Teasdale and Al Usher, who I worked with in my first startup experience while studying for my PhD. My PhD journey has been an intense but wonderful experience I'm happy to have shared with many friends and family.

Publications

During the course of this study, the following refereed journal and conference papers were published.

- Hinwood, D., Herath, D. and Goecke, R. *Towards the Design of a Human-Inspired Gripper for Textile Manipulation*. In Proceedings of the 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), pages 913–920. IEEE, August 2020.

Abbreviations and Shorthand Expressions

ANOVA	Analysis of Variance
DH	Denavit-Hartenberg
DH	Dependant Variable
DoA	Degrees of Actuation
DoF	Degrees of Freedom
CoM	Centre of Mass
CAD	Computer-Aided Design
CPI	Conservative Policy Iteration
EC	Environmentally Constrained
EE	End-Effector
F-GRASP	Refers to a fabric-specific anthropomorphic taxonomy defining grasp poses, derived from [Feix 15]
F-IHDM	Refers to a fabric-specific anthropomorphic taxonomy defining in-hand actions, derived from [Bullock 12]
FDM	Fusion Deposition Modelling
GRASP	Refers to an integrated research project funded by the European Commission
HT	Homogeneous Transformation
ID	Inverse Dynamics
IMU	Inertial Measurement Unit
IV	Independent Variable
IHDM	In-Hand Dexterous Manipulation
MEF	Maximum Entropy Framework
MDP	Markov Decision Process
MSBE	Mean-Squared Bellman Error
N	Newton
NN	Neural Network
KL	Kullback-Leibler

LED	Light Emitting Diode
LSTM	Long Short-Term Memory
PCA	Principal Component Analysis
PCB	Printed Circuit Board
PCC	Per Capita Consumption
PD	Proportional Derivative
PER	Prioritised Experience Replay
POMDP	Partially Observable Markov Decision Process
RL	Reinforcement Learning
RM-ANOVA	Repeated Measures Analysis of Variance
RNE	Recursive Newton-Euler
ROBEL	Robotics Benchmarks for Learning
SOA	state-of-the-art
TD	Temporal Difference
TCP	Tool-Centre-Point
ToF	Time-of-Flight
URDF	Unified Robot Description Format
VF	Virtual Finger

Summary of Notation

Kinematics

$T_{x,y,z}(i)$	A homogeneous transformation (HT) matrix indicating a translation of i metres along an axis.
$R_{x,y,z}(i)$	An HT matrix indicating a rotation along an axis (x,y,z) of i radians.
A_i	An HT matrix representing the compounding transformation of row i of the DH parameters.
A_w	A HT matrix representing a static transform to orient a structure relative to the world frame.
θ_i	A rotation about the z -axis in the DH parameters at row i .
d_i	A displacement along the z -axis in the DH parameters at row i .
a_i	A displacement along the x -axis in the DH parameters at row i .
α_i	A rotation about the x -axis in the DH parameters at row i .
q_i	The position of the i th actuator; units are in meters or radians.
J_i	A shorthand reference to an actuator within the mechanical system.
q	A vector representing the actuator positions.
$J(q)$	A function representing the Jacobian matrix of a serial manipulator.
τ_i	A value representing the force/torque applied by the i th actuator.
τ	A vector representing the actuator forces.
$f_{x,y,z}$	A value representing a force applied along an axis (x,y,z) .
$m_{x,y,z}$	A value representing a moment of force (torque) applied along an axis (x,y,z) .

Dynamics

\mathcal{L}	Lagrangian
\mathcal{T}	Kinetic Energy
\mathcal{U}	Potential Energy
\dot{q}	A vector representing the actuator velocities.
\ddot{q}	A vector representing the actuator accelerations.
$M(q)$	A function representing the joint-space inertia matrix.
$C(q, \dot{q})$	A function representing the Coriolis matrix.
$G(q)$	A function representing the gravity term.
P_i	The position of the centre of mass (CoM) of link i in the world frame.
\dot{P}_i	The velocity of the CoM of link i in the world frame.
p_i	The constant position of link i 's CoM relative to the i th row of the DH parameters.
m_i	The mass in kilograms of link i .
I_i	The constant inertia tensor about the CoM of link i .
ω_i	The rotational velocity of link i in the world frame.

Reinforcement Learning

s	A state
a	An action
r	A given reward
(s, a, s', r)	A transition tuple taken each step, consisting of a state, action, next state and given reward.
π_ϕ	A policy neural network with weights ϕ , also known as an actor in actor-critic (AC) algorithms.
Q_{θ_i}	A value function neural network with weights θ , also known as a critic in AC algorithms. In modern approaches there are usually multiple critic networks denoted with i .
$\bar{\pi}_\phi$	A target variant of the policy network used in AC algorithms.
\bar{Q}_{θ_i}	A target variant of the value function network used in AC algorithms
$\pi_\phi(s) = a$	An deterministic policy, generates an action from a state.
$\pi_\phi(a s) = \mathbb{P}_{\pi_\phi}[A = a S = s]$	A stochastic policy, generates a probability distribution over possible actions given a state.
$\mathcal{N}(m, \mu)$	Refers to a function generating a normal distribution with a mean of m and standard deviation of μ .
γ	Refers to the discount factor, a term usually between 0.9 and 0.99 which dictates the impact of possible future rewards.

Contents

Abstract	iii
Certificate of Authorship of Thesis	v
Acknowledgements	vii
Publications	ix
Abbreviations	xi
Summary of Notation	xiii
List of Figures	xxiii
List of Tables	xxvii
1 Introduction	1
1.1 Motivation	1
1.2 A Robotic Solution	5
1.3 Thesis Contributions	6
1.4 Chapter Outline	7
2 Literature Review	11
2.1 Humans - The Expert Manipulators	12
2.1.1 Defining Grasp Pose	13
2.1.2 Defining Human Dexterous Manipulation	16
2.1.3 Environmental Exploitation Grasping	17
2.1.4 Human-Inspired Fabric Manipulation	20

2.1.5	Concluding Remarks	21
2.2	Dexterous Robotic Manipulation	21
2.2.1	Robotic Grippers	21
2.2.2	Targeting Deformable Manipulation	28
2.2.3	Serial-Chain Manipulators	31
2.2.4	Prototyping novel manipulators with commercially available components	32
2.3	Reinforcement Learning	33
2.3.1	Introduction	33
2.3.2	Algorithms	35
2.3.3	Overview of Learned Robotic Skills	38
2.3.4	Enhancing Learning	45
2.3.5	Conclusion	50
2.4	Research Gaps	50
3	A Human-Inspired Investigation of Dexterous Fabric Manipulation	53
3.1	Introduction	53
3.2	Manipulator Selection	56
3.2.1	Gripper 1 - The CLOTHILDE Gripper	57
3.2.2	Gripper 2 - The CloPeMa Gripper	59
3.2.3	Gripper 3 - Three Fingered Underactuated Manipulator	60
3.2.4	Gripper 4 - The NAIIST M2S Openhand	61
3.2.5	Gripper 5 - CAM-Follower Mechanism	62
3.2.6	Gripper 6 - Three Fingered Anthropomorphic Manipulator	63
3.2.7	Gripper 7 - Prismatic Sliding Manipulator	64
3.2.8	Gripper Set 8 - Garment Traversing Manipulators	65
3.2.9	Final Remarks	67
3.3	Taxonomies for Anthropomorphic Manipulation	67
3.3.1	Addressing Grasp Poses	67
3.3.2	Addressing Dexterous Manipulation	70
3.3.3	Application Towards Fabric Manipulation	74
3.4	A Treatise on Anthropomorphically Defined Grippers	74
3.4.1	Grasping	75

3.4.2	Dexterous Skills within Fabric Manipulation	78
3.4.3	Applications	84
3.4.4	Technical Review of Surveyed Grippers	86
3.4.5	Open Areas to Address	89
3.5	Human-Centric User Study	91
3.5.1	Introduction	91
3.5.2	Method	92
3.5.3	Results	95
3.5.4	Discussion	101
3.6	A Refined Scope for a Unique Manipulator Solution	102
4	A Novel Manipulator for Fabric Manipulation	105
4.1	Conceptual Design	108
4.1.1	Gripper Sensors	110
4.2	System Model	113
4.2.1	Kinematic Analysis	113
4.2.2	Dynamic Analysis	122
4.3	System Integration and Assembly	126
4.3.1	Actuator Components	129
4.3.2	Electronic Components	132
4.3.3	Software	135
4.3.4	System Evaluation	137
4.3.5	Estimating Dynamics from CAD	139
4.4	Environmentally Constrained Grasping	143
4.5	Evaluation of Grasping Characteristics	146
4.5.1	Grasp Strength Validation	146
4.5.2	Holding Garments	151
4.6	Concluding Remarks	152
5	Novel Motor Control Skills for Grasping with Environmental Constraints	155
5.1	Introduction	156
5.1.1	Environmentally Constrained Grasping	156

5.1.2	Scope for Reinforcement Learning	158
5.2	Implementation	160
5.2.1	Learning Algorithms	160
5.2.2	Formulation	168
5.3	Results - Simulation	177
5.4	Results - Hardware Deployment	183
5.5	Discussion	189
5.5.1	The Resultant Model	189
5.5.2	Options for Further Exploration	190
5.5.3	Further Takeaways	191
5.6	Conclusion	192
6	Thesis Conclusion	195
6.1	Research Summary	195
6.1.1	Addressing the Research Questions	196
6.1.2	Limitations and Recommendations	198
6.2	Future Research Directions	199
6.3	Final Remarks	200
	Bibliography	203
A	Human Observation Exercise	225
A.1	Consent Form	225
A.2	Questionnaire	230
A.3	HRI Data and Information	231
A.3.1	Analysed duration data in human observation exercise	231
B	Mathematical Modelling of the Effector	235
B.0.1	Homogeneous Transformations	235
B.0.2	Properties of the Rotation Matrix	236
B.0.3	Gripper Kinematics	240
B.0.4	Dynamics Calculation and Validation	247
B.0.5	Simplified Tau Expressions	258

B.0.6	Grasp Force Estimation Raw Data	263
C	Gripper Development	265
C.1	Acknowledgement of Collaborative Contributions	265
C.2	List of Electronic Components	266
C.3	Microcontroller Communication	270
C.4	Dynamic Parameters Estimation	270
D	Reinforcement Learning Supplemental Materials	275
D.1	Arbitrary Grasping Environment Details	275
D.1.1	Learning Vector Details	275
D.1.2	Domain Randomisation Components	276
D.2	Reward Class Pseudocode	278

List of Figures

1.1	An example of clothing in landfill, sourced from Liu [Liu 22].	1
1.2	Worldwide production volume of all textile fibres (blue) alongside the subset of produced chemical (synthetic or man-made) textile fibres (black) from 1975 to 2018 (<i>in 1,000 metric tons</i>)	2
1.3	The double diamond design thinking framework, Herath and St-Onge [Herath 22].	9
2.1	Summary of the GRASP taxonomy [Feix 15].	15
2.2	Examples of grippers performing EC grasping.	19
2.3	Examples of alternative gripper designs.	23
2.4	Examples of grippers with technical augmentations to assist with grasping behaviour.	29
3.1	Gripper 1 [Donaire 20]	58
3.2	Gripper 2 [Petřík 15]	59
3.3	Gripper 3 [Koustoumpardis 14]	60
3.4	Gripper 4 [Von Drigalski 17b]	62
3.5	Gripper 5 [Koustoumpardis 17]	63
3.6	Gripper 6 [Ono 07]	64
3.7	Gripper 7 [Shibata 12]	65
3.8	Grippers 8.1 , 8.2 and 8.3 [Sahari 10]	66
3.9	A visualisation of VF and Opposition Types, Iberall [Iberall 97].	68
3.10	The human thumb in an abducted (left) or adducted (right) position [Feix 15].	69
3.11	The F-GRASP taxonomy, an adapted form of the GRASP taxonomy, derived from Feix <i>et al.</i> [Feix 15].	70
3.12	The original IHDM taxonomy of Bullock <i>et al.</i> [Bullock 12].	71

3.13	An expansion to scenarios 14 and 15 of the IHDM taxonomy (Figure 3.12), which describes the rotation and translation of within-hand manipulations.	72
3.14	The F-IHDM, an adapted form of the IHDM taxonomy, derived from Bullock <i>et al.</i> [Bullock 12].	73
3.15	A summary of the grasping characteristics of the surveyed devices.	77
3.16	Examples of two grasping techniques that exploit the environment.	79
3.17	Grippers 3 (<i>left</i>) and 1 (<i>right</i>) performing EC grasping with dexterous manipulation annotations and grasp pose tags.	80
3.18	Gripper 7 performing a unique biomimetic grasping technique, sourced from Shibata <i>et al.</i> [Shibata 16].	82
3.19	A summary of the dexterous skills from the discussed devices.	85
3.20	An example of the observation setup.	93
3.21	Histograms of the time-taken while performing several manipulation tasks under the three constraints.	96
3.22	Box-plots of the time-taken while performing several manipulation tasks under the three constraints.	97
3.23	A visualisation of participant responses to how difficult tasks under constraints were when compared to the unconstrained condition.	100
4.1	The lateral grasp in the poses Lateral Extension Grasp (LEG), left, and Lateral Flexion Grasp (LFG), right.	105
4.2	A rendering of the gripper prototype alongside the developed hardware platform.	106
4.3	The gripper with components labelled.	107
4.4	The iterations of the gripper across development.	109
4.5	Dimensions and sensing coordinate frame of the triaxial force sensor.	112
4.6	Visualisation of the DH parameters alongside a rendered gripper at a similar pose.	114
4.7	The range of motion of the built prototype.	115
4.8	Displacement of the triaxial force sensor when attached to the robotic manipulator.	117
4.9	A mechanical drawing of the gripper where <i>detailed view A</i> shows the position of q_2 (at $\frac{\pi}{2}$) causing a kinematic singularity.	121
4.10	An orthogonal engineering drawing of the developed gripper.	127
4.11	An exploded assembly diagram of the developed gripper.	128

4.12	The final gripper prototype attached to a Baxter robot.	129
4.13	Rendering of the prismatic rail component coupled to the stepper motor ($J0$).	130
4.14	Measured stall torque readings of a Dynamixel XM-430-350-R actuator.	131
4.15	An overview of the electronic architecture.	134
4.16	The custom ROS message structure for the gripper.	136
4.17	The gripper URDF model, visualised with RVIZ.	139
4.18	The example trajectory made to estimate the required actuator forces.	140
4.19	The forces required to execute the trajectory of Figure 4.18 under two orientations. Each row represents the gripper at a different orientation visualised on the right.	141
4.20	A visualisation of the static plate component demonstrating the variables for estimating a surface’s position when the gripper makes contact.	144
4.21	Calculation of level surface contact visualised with a y_{rot} and z_{rot} value of 40°	145
4.22	The initial EC grasping evaluation using pre-programmed trajectories [Hinwood 20].	145
4.23	The fabricated gripper with a loadcell attached to the plate.	147
4.24	Comparison of expected torque running through $J2$ while in the LEG configuration (visualised), compared to stall torque estimates of Figure 4.14.	149
4.25	Comparison of expected torque running through $J2$ while in the LFG configuration (visualised), compared to stall torque estimates of Figure 4.14.	150
4.26	The gripper holding various garments.	151
5.1	A participant in the user study conducted by Eppner <i>et al.</i> [Eppner 15].	156
5.2	Example scenarios that the RL problem will address by learning <i>arbitrary</i> grasping, showing the gripper pinning fabric to the table surface from a range of wrist orientations.	159
5.3	Network Structures used in TD3.	165
5.4	Network Structures used in SAC.	166
5.5	The learning environment in PyBullet with various surface textures.	170
5.6	The ROS framework that interfaces the simulation with the off-policy learning process.	172
5.7	The triaxial force sensor with its sensing coordinate frame and centre-point highlighted.	173
5.8	A visualisation of key components of the gripper learning setup.	175
5.9	The hours of training and steps taken while learning with TD3 under various seeds using hyperparameters from Table 5.1.	180

5.10	The hours of training and steps taken while learning with SAC under various seeds using hyperparameters from Table 5.2.	181
5.11	The average reward, along with the standard deviation visualised, is shown across the training seeds for each algorithm.	182
5.12	A plot of the exponential multiplier (m_t) applied to Fz from the real world sensor data given Fz lies between 0.4 and 1.44.	184
5.13	The target garments for EC grasping with the real-world gripper.	185
5.14	The gripper grasping a scarf using a TD3 policy trained only in the simulation.	187
A.1	A visualisation from the ANOVA analysis of Jamovi showing the degree of difficulty participants found pincer and lateral conditions compared to the unconstrained setting while grasping, folding and unfolding. The y-axis represents the difficulty magnitude responses on the form in Appendix A.2.	233
C.1	A comparative view of URDF and DH coordinate frames.	271

List of Tables

3.1	Friedman test with Durbin-Conover pairwise comparisons (generated from the Jamovi software).	98
3.1.1	Folding T-shirt	98
3.1.2	Sorting fabric	98
3.1.3	Sorting with folding	98
3.2	The mean, median, and mode information for the duration (seconds) of various tasks under different constraints. <i>UC</i> refers to unconstrained behaviour.	99
3.3	Paired Samples T-test with Wilcoxon rank evaluation - Participants response on grasp constraint difficulty.	100
4.1	Denavit-Hartenberg parameters	114
4.2	The byte message structure that communicates between the ROS host and microcontrollers.	135
4.3	Microcontroller evaluation of the gripper.	137
4.4	Communication latency evaluation of the gripper.	138
4.5	The dynamics parameters of the URDF and derived variants for the modelling process.	142
4.6	A comparison of the grippers discussed in Chapter 3 with the fabricated gripper.	153
5.1	The hyperparameters of applied TD3 algorithm	179
5.2	The hyperparameters of applied SAC algorithm	179
5.3	Results of hardware EC grasping using TD3.	186
A.1	The duration data (in seconds) of participants performing various tasks in the human observation exercise, highlighted blank yellow cells indicate errors made by a minority of participants.	231
A.2	Averages of the participant's times while folding a shirt.	231

A.3	Participant times while sorting clothing.	232
A.4	Participant times while folding and sorting clothing.	232
A.5	The recorded responses of the degree of difficulty participants found the pincer or lateral constraints compared to the unconstrained. This image is from the Jamovi software.	233
B.1	The measured and estimated grasp forces of the gripper.	263
C.1	The available destination and commands for the serial protocol.	270

Chapter 1

Introduction

1.1 Motivation

Environmental damage occurs at many steps in the fabric production and disposal cycle as Pensupa *et al.* [Pensupa 17] detail. Their overview of the topic provides an in-depth description of the devastating environmental impact of the clothing industry along with trends in sustainability. One can attribute the ecological damage of the clothing industry to both the physical waste and the chemical byproducts, primarily wastewater, generated during manufacturing. One can classify physical textile waste as either pre-consumer or post-consumer, with the former being the waste from production and the latter being disposal after use. Due to expanding consumption, levels of post-consumer waste have risen. For exam-



Figure 1.1: An example of clothing in landfill, sourced from Liu [Liu 22].

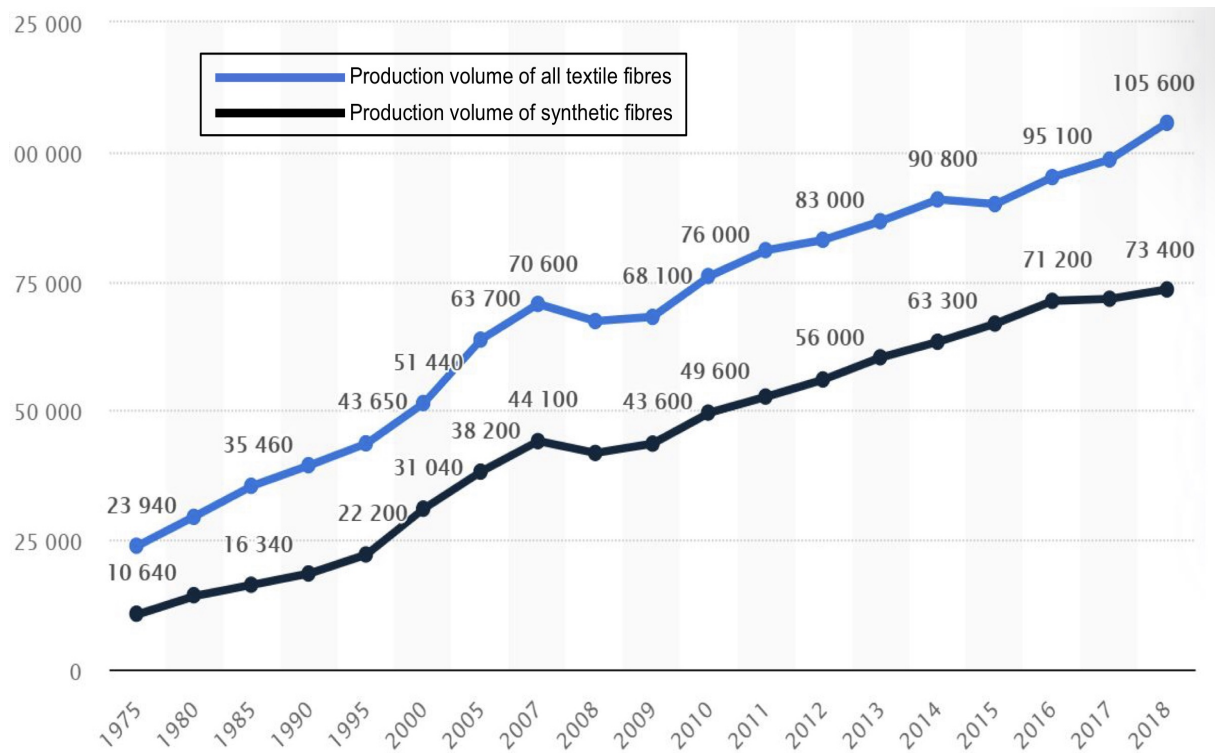


Figure 1.2: Worldwide production volume of all textile fibres (blue) alongside the subset of produced chemical (synthetic or man-made) textile fibres (black) from 1975 to 2018 (in 1,000 metric tons)¹.

ple, 13 million tons of clothing were sent to landfills in the United States of America (USA) in 2009, where 14.9% was recycled [Joung 13]. Per an Environmental Protection Agency (EPA) report [EPA 21], the mass of textiles sent to landfills increased to approximately 17 million tons in 2018.

Shui *et al.* [Shui 11] note a consistent increase in global per capita consumption (PCC) of clothing from 3.7kg in 1950 to 10.4kg in 2008. Also noted was that developed countries have been the primary contributor to most of the growth. However, in more recent decades, the PCC of developing nations has been growing at a significantly higher rate. Shui *et al.* [Shui 11] also observe an increased demand for cheaper synthetic fibres due to the resources required to manufacture natural fibres. Approximately 76 million tons of textiles were produced globally in 2010. This increased to 106 million tons in 2018 (Figure 1.2). Despite the organic components in fabrics, they are slow decomposing waste objects, taking up to twenty years depending on the composition to biodegrade [Pensupa 17]. Synthetic fibres are a significant contributor to increased decomposition time. Additionally, decomposing clothing in landfills can release toxic gases contributing to CO_2 emissions.

Each stage of a garment's commercial life-cycle presents an opportunity for possible polluting events [Clau-

¹This statistic comes from several IVC (Industrievereinigung Chemiefaser) chemical fiber industry updates. Source: <https://www.statista.com/statistics/263154/worldwide-production-volume-of-textile-fibers-since-1975/>.

dio 07]. Claudio also describes the ecological impact of textile manufacturing and the potential benefits of recycling. The production process of synthetic fibres can be an emission-intensive process requiring crude oil. The manufacturing of natural fibres, with cotton being one of the most popular, can also be environmentally damaging.

Clothing manufacturing can also damage the environment from the wastewater generated when garments undergo dyeing, rinsing or chemical refinement. The fast-fashion and clothing industry has produced approximately 20% of the world's industrial wastewater pollution, according to Kant [Kant 11]. Kant also describes the damaging impact of wastewater on the environment and infrastructure, including corrosion of irrigation frameworks, breakdown of drinking water supplies and evaporated chemicals affecting the air quality in populated areas.

Authors including Bick *et al.* [Bick 18] and Niinimäki *et al.* [Niinimäki 20] highlight how rising consumption and efficiency have lowered pricing and motivated a regular, trend-based consumer pattern, also known as 'fast-fashion'. The fast-fashion business model exacerbates the previously discussed environmental concerns and has led to more demand and unsustainable production measures, representing a critical environmental threat. Furthermore, Niinimäki *et al.* [Niinimäki 20] elaborate upon how the globalisation of the textile industry contributes to pollution inequalities and consequences. Many developing nations suffer the byproducts of manufacturing, while developed nations are often the primary beneficiaries of the produced garments.

Additionally, many developed nations often export their post-consumer textile waste to developing nations. However, this unsustainable paradigm is beginning to fail, with nations refusing to import textile waste for various reasons, such as protecting local economies. Fast-fashion also produces the issue of 'deadstock', a specific type of pre-consumer waste that consists of garments manufactured but unsold or unused. Usually, incineration disposes of this waste, but such practices destroy the substantial resources that went into manufacturing. Changing the present status quo of the textile industry is a multifaceted and complex endeavour involving modifications to both industry practices and consumer habits. Niinimäki *et al.* [Niinimäki 20] specifically state that long-term sustainability of the textile industry will require abandonment of the fast-fashion model. In the late 2010s and early 2020s, media outlets have reported on the devastating impact of textile waste, including the article of Besser [Besser 21], which details the impact of textile exports to Ghana. Approximately 40% of clothing received there is not wearable due to poor quality. The World Bank provided 9.5 million *EUR* to construct the Kpone landfill to address Ghana's rising textile waste problem. This landfill was full after five years of use. In 2019, the decomposing ma-

terial caused a methane bubble, which eventually lit and resulted in a fire for 11 months. Unfortunately, global demand has not stopped, and Ghana still receives approximately 15 million garments per week. Elsewhere, Siegle [Siegle 19] details the poor practices of the fast-fashion industry, including misrepresenting sustainability practices and poor working conditions in developing nations. Siegle specifically references the collapse of the Rana Plaza complex in Bangladesh, which killed over 1000 workers. The broader point is that textile manufacturing issues and poor industry practices are common, impacting the world, and require immediate attention.

Several proposed methods to combat the environmental impact of textiles in landfill exist. Zamani *et al.* [Zamani 15] present a case study on Swedish textile recycling practices, mentioning techniques including energy recovery (incineration), material reuse and chemical recycling. Incineration to create energy is a popular, conventional method that combusts textiles at high temperatures before performing energy recovery, such as using the heat produced to generate steam which powers generators. However, as outlined by Zamani *et al.* [Zamani 15], this process generates emissions and is considered a potential high global warming activity. Pensupa *et al.* [Pensupa 17] also discuss the emissions associated with incineration and detail how this process can create and release leachate.

Material reuse is another option to divert clothing from landfills, in which post-consumer waste is either resold or repurposed. Pensupa *et al.* [Pensupa 17] mention that such approaches exist, but repurposing occurs only with a minority of textile waste. Zamani *et al.* [Zamani 15] elaborate that material reuse usually only applies to niche markets and relies heavily on the quality of the collected material and the demand for reuse. Finally, chemical recycling processes are another possibility that can divert textile waste from landfills. Zamani *et al.* [Zamani 15] discuss the process of separating cellulose (an organic polymer found in natural fibres) from polyester with NMMO (N-Methylmorpholine N-oxide) solvent, and the processes surrounding polyester recycling. The first step in recycling polyester is to separate pure polyester garments from mixed or cellulose textile waste via manual labour. Pensupa *et al.* [Pensupa 17] expand on these techniques with an in-depth discussion surrounding various chemical-based fabric waste valorisation processes.

Several benefits are associated with these recycling processes. For example, Claudio [Claudio 07] discusses how using recycled cotton instead of virgin material can save up to 20,000 litres of water per kilogram of cotton grown. The review of Pensupa *et al.* [Pensupa 17] concludes with statements regarding the current methods for textile recycling while highlighting barriers. The authors note the extensive labour required to separate fabric waste by size, composition and fibre type parameters, a task

made more difficult with traditional waste streams [Wang 10]. Zamani *et al.* [Zamani 15] note that the range of garment compositions to be recycled will influence the structure of any recycling endeavour. One example of an existing recycling project by Heikkilä *et al.* [Heikkilä 19] separates four categories of textile waste with near-infrared (NIR) technology at 90% accuracy. However, a collection and sorting system will be an essential component for large-scale recycling endeavours. The sheer volume of textile waste generated makes manual labour intensive, expensive and economically impractical for a sorting module. However, robots could perform some of the manual labour involved in sorting textiles to reduce costs and overcome the discussed limitations.

1.2 A Robotic Solution

As noted, recycling of diverse textile waste streams remains limited. While several commercial endeavours around the globe have attempted to implement sorting systems, their capabilities are usually limited to specific waste streams or polymers. Zamani *et al.* [Zamani 15] recognise that textile waste's diverse compositions and colours can limit commercial applications. Wang [Wang 10] describes how targeting specific materials is a preferred commercial structure, as businesses can target specialised processes or recycling outcomes and discusses how certain endeavours aim for particular waste streams, such as uniforms for sport or education institutes. Alternatively, some processes target alternative textile items such as carpets. However, such endeavours are still limited by sorting capabilities.

While exploring the issue of textile waste, discussions with Australian textile recycling companies revealed the difficulties commercial endeavours experience with diverse waste streams. They detail how using human labour to sort textile waste is costly and puts labourers at risk of repetitive strain injuries. Additionally, humans can have an adverse response to specific items within the textile waste, such as undergarments, which are considered unhygienic. These factors have resulted in companies targeting only the most profitable textile waste streams. Part of the recycling process outlined by these companies involved using human labour to sort clothing into broad categories based on the type of garment or colour. While more complex actions, such as removing buttons, are present in the pipeline, addressing this first sorting step with robotic labour could empower recyclers to pursue a broader range of waste streams for recycling. The long-term vision of this research is to enable robots to perform simple but laborious and repetitive tasks in the recycling pipeline, working alongside humans and refining the sorting process, to bring textile waste management closer to a circular economic model.

As elaborated in subsequent chapters, robots are currently limited in their fabric manipulation capabilities. Therefore, an investigation and development process of a novel end-effector designed for deformable manipulation has been undertaken. Previously observed end-effectors designed for fabric manipulation do not generalise to a wide range of applications, nor do they capture a complete set of manipulation skills needed to effectively grasp and sort clothing in a wide range of environments. As humans presently remain the most effective manipulators of fabrics and deformable objects, a set of skills and features are derived from anthropomorphic and human-centric studies to inform the development of this proposed robotic end-effector. This generalised solution could contribute to autonomous fabric waste sorting applications. Finally, as part of developing the robotic gripper, state-of-the-art reinforcement learning (RL) algorithms train the device to perform grasps that leverage environmental constraints, which is currently a novel challenge in the scope of fabric manipulation.

1.3 Thesis Contributions

- The first contribution develops anthropomorphic hand-centric taxonomies that describe fabric manipulation primitives. Currently, many grippers have been developed to address aspects of deformable manipulation. However, these devices use heuristic observations of human behaviour to develop solutions or create devices with complex design components. In order to address these limitations, this thesis creates fabric-specific derivatives of state-of-the-art hand-centric taxonomies to qualify the grasp pose and dexterous capabilities. These custom taxonomies are novel as they are the first example of hand-centric taxonomies designed to qualify manipulation primitives targeting fabric manipulation.
- The second contribution discovers novel insights into gripper design for generalised solutions targeting deformable manipulation. Prior grippers developed for fabric manipulation are analysed under an anthropomorphic lens using the taxonomies created in the previous contribution. This study is the first example of an investigation that surveys hand-centric features of grippers targeting fabric manipulation. Observations from this study are validated by a user study examining how different grasp constraints impact humans' ability to grasp and manipulate clothing.
- The third contribution models and builds a novel gripper for textile pick-and-place applications. This thesis conceives of a novel gripper that fulfils the requirements of a generalised pick-and-place grasping solution for textile waste by building upon the observations discovered in the second con-

tribution. The development includes using established rigid-body kinematic and dynamic analyses to understand the system's physical characteristics alongside the development of the electronics and integration into a robotics framework. After this contribution, a unique gripper is developed compared to the existing state-of-the-art and becomes a platform for the reinforcement learning undertaken as the fourth and final contribution.

- The final contribution develops a reinforcement learning reward schema inspired by human observations that enables reinforcement learning algorithms to learn grasping motions that exploit the environment. Previous research projects have investigated how humans leverage the environment while grasping flattened materials. This contribution presents the first example of a data-driven approach towards grasping motions that leverage the environment under a fabric manipulation context. The work undertaken involves the development of a simulated reinforcement learning environment with the gripper developed in the previous contribution and an evaluation of trained policies on the hardware platform.

1.4 Chapter Outline

Chapter 2 contains a literature review detailing the theory, techniques and inspirations that inform the research. The literature review surveys topics including anthropomorphic manipulation, autonomous fabric handling, robotic grippers, and reinforcement learning. Investigation of human-inspired manipulation requires reference to taxonomies derived from observed human manipulation behaviour, using descriptors that can classify both grasp pose and dexterous manipulation. Therefore, a focus on anthropomorphic classification schemes and their applications is presented. The survey also discusses how anthropomorphic taxonomies can inspire novel gripper designs or desired robot behaviour.

The next section of the literature review discusses robot manipulators, the design of devices targeting fabric manipulation, challenges within manipulating clothing, and various trends in dexterous manipulation. Additionally, details of traditional rigid body kinematics, kinostatics, and dynamic modelling techniques are presented, recognising that the proposed novel gripper in Chapter 4 is a serial-link manipulator. These modelling methods allow for the evaluation of forces exerted by actuators and their relationship with the positions, velocities, accelerations and wrenches of the tool-centre-point (TCP). Finally, data-driven techniques for learning intelligent robotic behaviour are discussed. To this end, prior work surrounding reinforcement learning techniques and applications within robotics is presented. The

chapter highlights state-of-the-art (SOA) methods and how they can apply in this project's research, which performs the learning of dexterous manipulation skills.

A recent general-purpose survey of robotic grasping [Babin 21] discusses current challenges in gripper development, including design approaches and applications, noting that modern development of robotic systems requires a multidisciplinary approach involving fabrication/design principles and various engineering disciplines, including electronics, software and mechanics. Taxonomies are also often utilised as a mechanism to qualify parameters, such as anthropomorphic grasp pose [Feix 15], contact relationships [Borràs 20], or manipulation primitives [Heinemann 15]. In designing robot manipulators, Babin and Gosselin [Babin 21] discuss the complexity of defining manipulation skill-set and scope, noting some approaches use taxonomies to define capabilities of solution-specific manipulators while others attempt to formulate generalised solutions to navigate a wide range of manipulation tasks. At a high level, it is concluded that the conundrum involves determining whether the desired task informs the design or whether the inverse relation is true. This discussion can also inform sensor choice and placement.

While surveying the literature, a human-inspired theme was found to be common amongst manipulators developed to manipulate textiles. This theme could be expressed as anthropomorphic features within manipulators, or the construction of manipulators that replicate human behaviours. Combining the human-inspired theme of previous devices aimed at fabric manipulation and the recommended multidisciplinary approach of Babin and Gosselin [Babin 21], this thesis follows a multidisciplinary approach to formulate a novel gripper for textile waste sorting. Each chapter presents an investigation into gripper design and textile manipulation under a specific discipline which contributes to the formulation, development and training of a novel manipulator. Additionally, the formulation and development of the gripper, which occurs across Chapters 3 and 4, follows the **Design Thinking** approach [Lewrick 18].

Design thinking, at a high level, is a cognitive process for finding solutions to indeterminate issues by carefully studying key stakeholders and working through an iterative design process. Peng discusses the design thinking conceptual framework in the textbook by Herath and St-Onge [Herath 22], describing its origins and highlighting various interpretations. The approach outlined by Peng follows a version of design thinking called the *Double Diamond* approach (Figure 1.3), which Chapters 3 and 4 follow to deliver the novel gripper.

The double diamond approach uses the four steps of *discover*, *define*, *develop*, and *deliver*. First, the *discover* step identifies a core problem set that requires addressing. The next step, *define*, identifies areas to improve by gathering insights about present limitations. One can iterate through the steps of

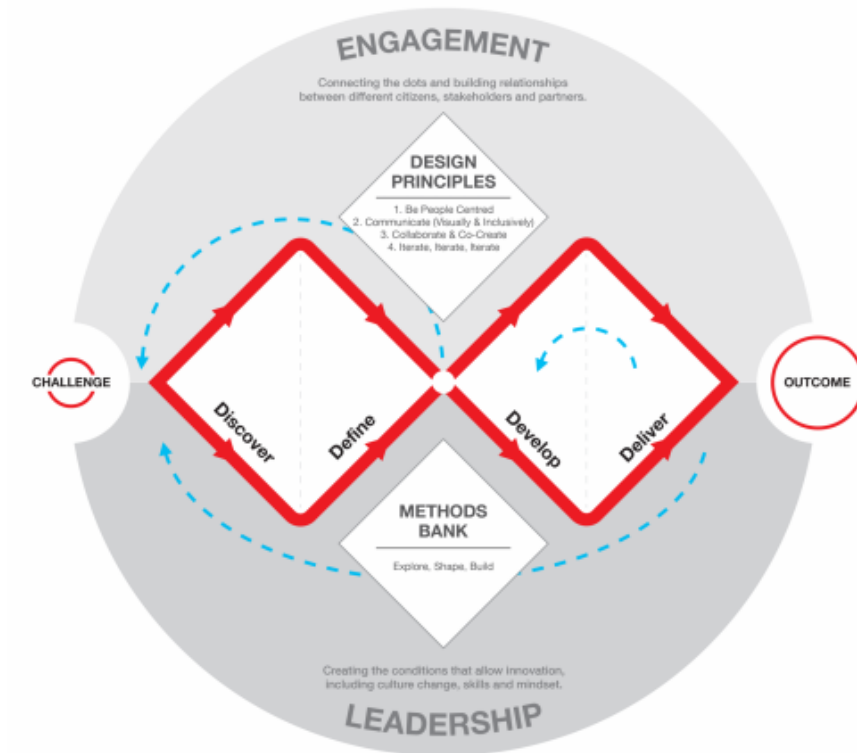


Figure 1.3: The double diamond design thinking framework, Herath and St-Onge [Herath 22].

discover and *define* multiple times, as visualised in Figure 1.3, while establishing a clear problem scope. The third stage, *develop*, consists of the two sub-steps of *ideate* and *prototype*. Ideation occurs by formulating solutions based on the insights gained through the previous steps. The prototyping stage develops a clearly defined solution from the conceptual stages of *discover*, *define* and *ideate*. In addition, an evaluation of developed prototypes takes place to determine if the proposed solution fulfils an ideated vision or requires improvement. Prototyping is an iterative process exploring a solution's risks and opportunities while refining the design. The final step of *deliver* brings the prototype to an application organisation with further validation occurring, which confirms that the constructed prototype fits the intended organisation's requirements. Figure 1.3 highlights that further iteration can occur across the complete process of *discover*, *define*, *develop*, and *deliver* if required.

Within the double diamond approach, the literature review performs the step of *discover* by identifying present limitations and unexplored research avenues. The first research-oriented project, detailed in Chapter 3, performs the steps of *define* and *ideate* to formulate a novel gripper design by surveying existing custom-designed grippers under an anthropomorphic manipulation lens. Such an approach investigates the robotic manipulation of fabric from a neuroscientific and psychological viewpoint. Topics

discussed include the applications, design inspirations and grasping strategies. The core contribution of Chapter 3 uses anthropomorphic hand-centric classification schemes to identify previously unknown gaps from existing manipulators, resulting in a novel conceptual robot gripper based on a human grasp configuration. In contrast, previous research has created manipulators inspired by specific human behaviours, desired garment configurations or specific manipulation primitives. The novel gaps and conceptual design derived from the anthropomorphic survey lay the groundwork for the research performed in Chapters 4 and 5. A user study also validates that the conceptual gripper design is sufficient for generalised textile waste sorting.

Chapter 4 continues the double diamond approach by performing the *prototype* step, using classical mechanics, electronics engineering and software development practices to model and fabricate the novel gripper. This research uses established practices to model the gripper's kinematics, kinostatics and dynamics, describing the system's motion and physical characteristics. Following these details, a description of the fabrication process, control mechanisms, integrated sensors, and ROS architecture is presented. Finally, Chapter 4 concludes with an evaluation of the gripper's mechanical capabilities, including grip strength and basic grasping actions. In terms of the double diamond methodology, this chapter concludes by executing the final *deliver* step, completing the design thinking process and producing a research prototype that is ready for a data-driven investigation.

The investigation of Chapter 5 enhances the produced gripper by applying reinforcement learning techniques to teach dexterous behaviour that leverages the environment to assist with grasping flattened materials. Such a skill is commonly associated with the exemplary manipulation capabilities of human beings [Eppner 15]. Continuing the human-inspired theme of Chapter 4, the contribution of Chapter 5 studies fabric manipulation from a machine learning perspective, which formulates a reward signal based on observed human behaviour from the literature and then trains two off-policy RL algorithms to execute these contact-rich grasping motions on the novel gripper. Such an approach is novel to grasping flattened materials and executing environmentally constrained grasping. Following the research chapters, the final discussion and concluding remarks outline the novelty and further work to pursue.

Chapter 2

Literature Review

Sanchez *et al.* [Sanchez 18] discuss robotic manipulation of deformable materials and highlight the complexity involved. Their review encompasses manipulating and sensing fabric, rope and quasi-deformable materials, including paper and sponges. A range of technical challenges for robotic systems become apparent when attempting to perform seemingly simple fabric-based chores such as folding laundry or placing garments on coat hangers. Clothing exhibits little or no compression strength and will respond with non-linear mechanical behaviour to applied forces. Such features result in garments displaying a near-infinite range of physical states for a single item. In addition, deformations such as wrinkles or creases usually store low elastic energy [Jiménez 17], meaning clothing will generally not spontaneously revert to previous states or configurations. Finally, differing fabric compositions or weaves result in garments displaying various mechanical behaviours and unique appearances, further complicating visual interpretation and manipulation. Within the broad topic of robotic textile manipulation, this thesis focuses on formulating, developing and training a novel robot gripper that sorts garments in diverse textile waste streams. Such research requires understanding related topics, research gaps and possible methodologies, which this chapter addresses with a literature review. A common theme through the literature was the way in which humans are considered the most advanced manipulators. Therefore, the developed gripper and dexterous skills formulated throughout the research chapters take inspiration from observed human-centric capabilities for autonomous fabric handling.

2.1 Humans - The Expert Manipulators

Humans are exemplary and adaptable physical manipulators of the world [Feix 14, Huang 21, Eppner 15]. Therefore, a significant amount of previous research investigates human manipulation behaviour. These efforts include development of novel taxonomies that use neuroscientific definitions to describe anthropomorphic aspects [Cutkosky 89, Feix 15, Kang 92]. Other research looks to observational studies of human behaviour while manipulating objects [Kazemi 14, Heinemann 15, Sarantopoulos 18]. These approaches try to understand human manipulation's behaviours, kinematics, constraints and capabilities to assist the transference of dexterous skills to robotic modules. Applying skills to robots can involve incorporating observations into motion planning approaches, designing anthropomorphic manipulators, or replicating dexterous skills in autonomous systems.

Understanding human-like manipulation in a robotic context has been a significant investigation area encompassing studies that examine human grasp poses or manipulation behaviour. Such research commonly creates taxonomies that describe certain aspects of anthropomorphic manipulation. Taxonomies can take various forms, including tree data structures, hierarchies or matrices. Taxonomies can also define manipulation aspects at a high level or target specific behaviours. For example, Heinemann *et al.* [Heinemann 15] present a binary-tree-based taxonomy aiming at a specific set of manipulation primitives involved in grasping behaviour that leverages the environment. While describing manipulation, discussed aspects can either be hand-centric or object-centric. Hand-centric aspects focus on the manipulator pose and motion to qualify the grasp or manipulation primitive. In contrast, object-centric parameters utilise information from the manipulation target or task context [Cutkosky 89].

Object-centric definitions can assist when defining manipulation strategies while considering the target object's state and desired configuration. An object-centric taxonomy [Borràs 20] describes grasps in fabric manipulation based on the intrinsic and extrinsic virtual fingers¹ present within a manipulation. Borràs *et al.* built a comprehensive framework that describes clothing configurations based on environmental and manipulation interactions by surveying commercial and custom grippers for fabric manipulation. They discuss how point-to-point grasping remains common throughout grippers targeting fabric manipulation. Additionally, they highlight the environment surface's significant role while discussing various fabric manipulation scenarios. They use their framework to define manipulation primitives within fabric handling tasks to establish gripper designs and scope manipulation requirements. While this object-centric approach presents insights into fabric manipulation, no hand-centric investiga-

¹See Section 2.1.1 for a description of Virtual Fingers.

tions into the same topic have occurred, despite numerous grippers within this context inspired by the human hand [Von Drigalski 17b, Le 13, Koustoumpardis 14, Koustoumpardis 17].

Hand-centric taxonomies make descriptive comparisons to the human hand for the pose and motion of robotic manipulators. These descriptors can then describe actions or behaviours taken to perform grasping and dexterous skills. Using anthropomorphic hand-centric analysis techniques can also inspire designs, define functionality and inform system scope [Feix 15].

2.1.1 Defining Grasp Pose

Describing grasp configurations of a human hand in the context of robotic end-effectors has been a difficult task, as illustrated in various studies. Feix *et al.* [Feix 15] define a grasp as “every static hand posture with which an object can be held securely with one hand, irrespective of the hand orientation”. An early benchmark of human grasping behaviour in a robotic context [Cutkosky 89] is referenced regularly throughout grasping studies, and still influences modern anthropomorphic grasping taxonomies. Cutkosky [Cutkosky 89] also introduced concepts from the field of neuroscience, comprehensively discussed in the articles of Iberall *et al.* [Iberall 86, Arbib 85, Iberall 97]. These fundamental principles exist in the modern SOA (state-of-the-art) grasp classifiers, including the research of Feix *et al.* [Feix 15].

The starting feature to discuss is the power or precision classification of a grasp, referred to as the grasp type. This notion was initially discussed by Napier [Napier 56], who describe how all prehensile interactions can require power or precision features. Power grasps are static poses that rigidly hold an object, and motion must come from the arm or wrist. In contrast, precision grasps are seen as weaker grips which retain the ability to perform intrinsic dexterous manipulation. Cutkosky [Cutkosky 89] initially uses this descriptor to broadly separate grasp configurations.

However, classifying grasp types as this binary descriptor can be limiting. Thus, research projects such as Kamakura *et al.* [Kamakura 80] discuss an additional intermediate state for grasp types. This addition was necessary because certain grasp poses can feature elements of both power and precision in similar proportions [Feix 15]. An example of such a grasp would be the lateral grasp [Iberall 97], which has been classified as a power grasp by Cutkosky [Cutkosky 89] and Kang and Ikeuchi [Kang 92]. In contrast, others have described this pose as a precision grasp [Tyldesley 96].

Another concept discussed in the research of Iberall [Iberall 97] is ‘virtual fingers’ (VF). Originally described as an abstract representation of a collection of individual fingers and hand surfaces applying an oppositional force, any fingers that form a single body in a grasp merge into a single VF. This principle

assists in defining prehension, where a prehensile grasp is any interaction involving more than a single virtual finger. Non-prehensile refers to only a single finger being present in the manipulation action. Feix *et al.* [Feix 15] refine the definition of prehensile as an interaction that a single contact point cannot accurately represent. Also discussed in Iberall's research is the opposition type parameter. This feature describes the force directions within a grip with the options of palm opposition (grasping forces applied in a parallel direction to the palm), pad opposition (grasping forces applied in a perpendicular direction to the palm) or side opposition (grasping forces applied in a transverse direction to the palm). Further visualisation and description of these features are provided in Chapter 3 (See Fig. 3.9).

Grasp taxonomies exist in various forms. For example, Cutkosky [Cutkosky 89] utilised a binary tree structure that initially classifies grasps as power or precision, then further refines grips by prehension and target object geometry. This binary tree taxonomy starts by classifying grips with hand-centric definitions before refining the grasp with object-centric descriptors. A wide range of research projects has built upon this topic describing various grasps and applications. A more recent SOA taxonomy for anthropomorphic grasp comparison is the GRASP taxonomy [Feix 15].

The GRASP taxonomy, visualised in Figure 2.1, involved a comprehensive overview of the literature detailing anthropomorphic grasping definitions. The name of the GRASP taxonomy comes from the European project that funded the research. The GRASP taxonomy defines thirty-three grasps and sorts the observed grips into seventeen categories. Parameters such as grasp orientation, literature frequency, and contact points informed the taxonomy development process. The classifying architecture of the GRASP taxonomy utilised the four parameters of opposition types, grasp type, thumb adduction/abduction and virtual fingers for classification. The thumb adduction/abduction descriptor is a binary parameter that refers to the rotation of the thumb. For example, the colloquial hand symbol of a 'thumbs up' holds the thumb in an adducted position. Therefore, an abducted pose occurs when the base rotation is similar to forming a closed fist. The GRASP taxonomy is matrix structured with cells holding various grasps, their location within the matrix determined by their hand-centric characteristics.

While comprehensive, the GRASP taxonomy does not account for object-centric parameters or variations in contact. Thus several grasps described in the literature can populate a single cell within the matrix structure as shown in Figure 2.1. This taxonomy also omits uncommon grasp configurations, which results in some cells being empty. For example, Feix *et al.* [Feix 15] describe a grasp configuration holding a cigarette that is not present in Figure 2.1. Feix and colleagues also argue that such grasps are generally unnecessary to describe when discussing manipulation and easily accommodated if

		Power					Intermediate			Precision					
		Palm		Pad			Side			Pad			Side		
Opp:	VF:	3-5	2-5	2	2-3	2-4	2-5	2	3	3-4	2	2-3	2-4	2-5	3
Thumb Abducted			1: Large Diameter 2: Small Diameter 3: Medium Wrap 10: Power Disk 11: Power Sphere 	31: Ring 	28: Sphere Finger 	18: Extension Type 	19: Distal Type 	23: Adduction Grip 		21: Tripod Variation 	9: Palmar Pinch 	8: Prismatic 2 Finger 	7: Prismatic 3 Finger 	6: Prismatic 4 Finger 	20: Writing Tripod
		17: Index Finger Extension 	4: Adducted Thumb 					16: Lateral 	25: Lateral Tripod 						
Thumb Adducted			5: Light Tool 												
			15: Fixed Hook 				29: Stick 							22: Parallel Extension 	
			30: Palmar 				32: Ventral 								

Figure 2.1: Summary of the GRASP taxonomy [Feix 15].

required. The taxonomy structure also remains adaptable to develop task-specific classification schemes where one could optimise the scope of hand configurations and level of granularity. While possible gaps remain, the GRASP taxonomy presents a thorough survey of human grasp poses and provides a comprehensive grasp classification scheme.

An early version of the GRASP taxonomy [Feix 09] was used by Bullock *et al.* [Bullock 13, Bullock 15] to label observed grasps from a video dataset constructed by recording machinists and house-cleaners. Furthermore, the GRASP taxonomy also acts as a foundational framework that subsequent studies have used as an inspiration or baseline comparison for building their own classification architectures. Examples include the taxonomies of Abbasi *et al.* [Abbasi 16], Stival *et al.* [Stival 19], and Arapi *et al.* [Arapi 21], all of which were both influenced by and utilised the GRASP taxonomy. However, the more significant point in listing these examples is to demonstrate the general acceptance within the scientific community that the GRASP taxonomy of Feix *et al.* [Feix 15] is broadly considered a SOA

taxonomy for describing anthropomorphic grasp poses.

2.1.2 Defining Human Dexterous Manipulation

While the GRASP taxonomy presents a sufficient framework for describing anthropomorphic grasp poses, it cannot represent behaviour related to dexterous manipulation such as intrinsic manipulation, manipulator motion and object contact interactions. It can also be challenging to describe grasping and dexterous manipulation under a single classification architecture. Thus taxonomy architectures that can describe such behaviour are also discussed. There are many challenges in classifying manipulation behaviour, as task context and required manipulation skills vary. For example, the taxonomy of Heinemann *et al.* [Heinemann 15] describe actions such as reach, slide or flip, among others, referring to manipulation strategies that exploit the environment.

However, requirements exist for a more generalised framework targeting dexterous manipulation that can broadly characterise physical behaviour while considering contact, motion, and intrinsic actions. The research of Bullock *et al.* [Bullock 12] presents a taxonomy that describes human dexterous manipulation behaviours as a binary tree based on the parameters of contact, prehension, motion, within-hand motion and motion at contact. For future reference, the expression In-Hand Dexterous Manipulation (IHDM) taxonomy refers to their classification scheme. Unlike the GRASP taxonomy, which is purely hand-centric, the IHDM taxonomy accounts for additional information within an interaction regarding the object, environment and motion at the wrist. This taxonomy also applies to the context of robotics, and also presents an additional scheme to classify the direction and rotation of in-hand manipulation. Bullock and colleagues suggest pairing the IHDM taxonomy with a grasp taxonomy to classify dexterous manipulation and grasp behaviour simultaneously. Nakamura *et al.* [Nakamura 17] took this approach by using the GRASP and IHDM taxonomies alongside an additional medical taxonomy to capture a video dataset of manipulations occurring in a convenience store.

The IHDM taxonomy is applied explicitly in Chapter 3 to describe actions taken by robotic manipulators towards deformable manipulation. A pertinent observation about this taxonomy is that the structure and descriptors are generalised and abstract, and not specific to any particular gripper design [Liu 14]. However, Liu *et al.* [Liu 14] take inspiration from how the IHDM taxonomy defines motion to encode additional direction information into observed everyday activities alongside a more traditional manipulation primitive classification scheme.

2.1.3 Environmental Exploitation Grasping

Eppner *et al.* [Eppner 15] comprehensively surveyed the topic of grasping while exploiting environmental constraints and discussed the associated benefits. Others have noted that simplified effectors can be capable of grasping objects from a level surface [Xu 09, Odhner 12, Yoshimi 12]. These studies utilise a non-prehensile wiping motion from a compliant finger inward towards a grasp. The described wiping motion approximates how humans exploit environmental constraints with their fingertips. Eppner *et al.* [Eppner 15] also discuss shape adaptation, the conformity of a manipulator to a static grasping target. As uncertainty exists in computer vision, tactile sensing, and actuation modules, shape adaption presents one possible solution to these intrinsic complexities. Grippers that utilise soft robotic components or tendon-based mechanisms commonly perform shape adaption, an example being the iHY gripper presented by Odhner *et al.* [Odhner 14]. The benefit of overcoming intrinsic uncertainties with shape adaptation can apply to environmental exploitation grasping, where the approach can account for unknown parameters between the object and the environment surface.

An earlier article [Eppner 13] specifically demonstrated that methods with simplified vision components could use shape adaptability to account for the finer geometric details of an object and the environment. Eppner *et al.* [Eppner 15] also develop several environmentally constrained (EC) grasping strategies for grasping static objects from a surface with a Barrett robot hand, before using these grasping strategies to develop an anthropomorphic, pneumatic gripper with the explicit requirement of grasping static objects while leveraging environmental constraints. Dafle *et al.* [Dafle 14] introduce extrinsic dexterity, describing this concept as utilising external forces such as gravity or the environment to enhance in-hand manipulation, and demonstrate how simple grippers using these external factors are capable of a range of in-hand dexterous skills.

EC grasping is also considered anthropomorphic. For example, Eppner *et al.* [Eppner 15] performed a small human-centric analysis in which participants grasped objects from a flat surface. Participants were either unimpaired, or their vision was limited with a set of lenses. Participants resorted to environmentally exploitative grasps more often when impaired, indicating a reliance on such approaches when visual data streams were limited. Grasping duration was also increased when vision was impaired. Other authors have also considered human-centric interpretations of environmental grasping. For example, Puhlmann *et al.* [Puhlmann 16] built upon the research of Heinemann *et al.* [Heinemann 15], further analysing human behaviour to construct a framework for robot grasping while interacting with the environment. In that study, subjects participated in an observation exercise in a similar manner to Eppner *et*

al. [Eppner 15]. However, another constraint where participants could not use the environment was also applied. Participants saw a significant deterioration in grasp performance from the presented observation exercise when instructed not to use the environment.

Sarantopoulos and Doulgeri [Sarantopoulos 18] also present human-inspired grasping strategies that explicitly target environmental exploitation. An assumption made in this research was that both the robot gripper and manipulator were compliant mechanical devices. Similarly, Kazemi *et al.* [Kazemi 14] developed human-inspired environmental exploitation grasping strategies. Their observational study demonstrates how humans naturally gravitate towards environmental exploitation when permitted, especially when visually constrained. Kazemi *et al.* [Kazemi 14] then present a series of grasping strategies based on compliant actuation mechanisms for the grasping of relatively minute objects when compared to the robotic manipulator. Alternatively, Santina *et al.* [Della Santina 17] present an article that studies possible kinematic, synergistic strategies when performing environmental exploitation grasping. They observed hand pose behaviour in pre-shaping and interaction stages of grasping from human subjects. Like Eppner *et al.* [Eppner 15], Santina *et al.* [Della Santina 17] also noticed an increase in grasping duration, along with an increase in force applied to the environment when impeded. In their case, the impeded condition limited the tactile sensing of the participant's fingertip.

Many grippers contain passive components to account for the environment while grasping, especially manipulators that target fabric. Examples include the tendon-driven gripper of Koustoumpardis *et al.* [Koustoumpardis 14] or the industrial CloPeMa manipulator with variable stiffness control by Le *et al.* [Le 13]. Some generalised grippers perform EC grasping with soft components such as the pneumatic RBO hand by Eppner *et al.* [Eppner 15]. Commercial devices used by Sarantopoulos and Doulgeri [Sarantopoulos 18]² use intrinsic compliance within the joints or a joint admittance controller to perform EC grasping.

The discussed research projects indicate that compliant elements, either as internal control mechanisms or physical characteristics, are required for any manipulator that exploits the environment while grasping. However, modelling such an interaction can be complex, as demonstrated by Salvietti *et al.* [Salvietti 15]. They present a mathematical framework for modelling compliant hands and arms while interacting with the environment to understand the interaction between the hand, object and environment. Applications of the mathematical framework can evaluate arm/hand capabilities and infer design features. In a recent comprehensive review of robotic grippers, Babin and Gosselin [Babin 21]

²Manipulators utilised included the Barrett Hand BH8-282, Shadow Hand Lite and Shadow Smart Grasping System.

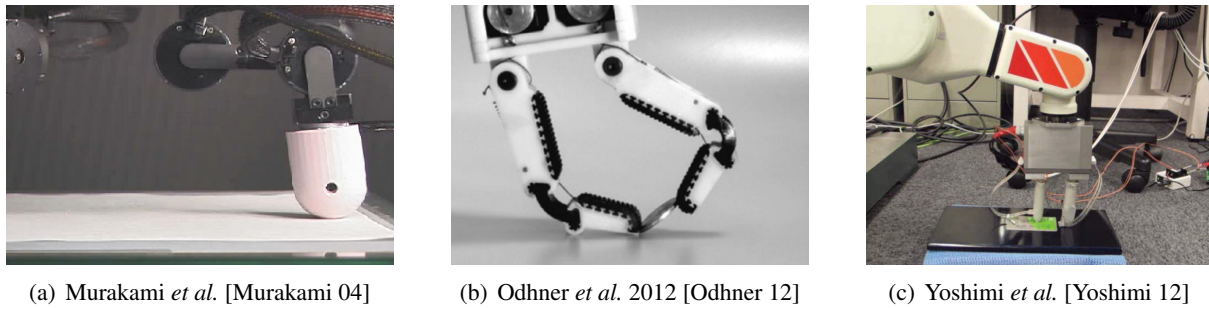


Figure 2.2: Examples of grippers performing EC grasping.

note the difficulties involved in environmental exploitation. They suggest using soft or flexible robot components to navigate this complex interaction.

Aside from human grasping techniques, physical features of the human hand have also inspired gripper features when performing EC grasping. Specifically, Murakami *et al.* [Murakami 04] present a soft fingertip with a hard nail to assist in manipulating flattened paper on a surface. They showed how such a design allowed for a dynamic friction adjustment within the interaction without significantly altering the applied normal force against the environment. Fingernail-like appendages can also be helpful when transitioning from an environment contact to a flat grasping target. The nail is inserted between the environment surface and the target to assist the grasp in such a situation. This approach is analogous to the method utilised by Yoshimi *et al.* [Yoshimi 12] while picking up a credit card. A similar approach was also seen by Odhner *et al.* [Odhner 12], which targeted a flip and pinch task using the Yale OpenHand T42 gripper. Odhner *et al.* [Odhner 12] performed a grasp exploiting the environment to obtain a precision grasp on a coin. While attempting to catch step points where the environment and coin are in contact, the fingertip adopted a design with ridges mimicking a fingernail's sharp contact features, which provided a wide range of contact normals from the side, improving the frictional contact with the object while grasping. Figure 2.2 visualises examples of the discussed grippers.

Grasps that navigate or leverage the environment are a set of contact-rich manipulation primitives. Elguea-Aguinaco *et al.* [Elguea-Aguinaco 23] specifically mention how contact-rich manipulation is a critical component of the changes introduced by Industry 4.0. Broader manipulation and grasping surveys highlight the importance of collision-rich grasping and extrinsic dexterity [Babin 21, Newbury 23, Elguea-Aguinaco 23, Suomalainen 22]. Furthermore, exploiting the environment while grasping is a fundamental skill for any manipulator targeting fabric manipulation, which battles uncertainty and provides a robust grasping methodology. The human-centric studies and gripper designs discussed in this section reinforce these statements. Several grippers designed for fabric manipulation target this capabil-

ity [Donaire 20, Le 13, Koustoumpardis 14]. Thus, any system developed for generalised fabric handling will likely need such a capability.

2.1.4 Human-Inspired Fabric Manipulation

Human manipulation behaviour and definitions have inspired gripper designs and strategies for fabric manipulation. Research projects developing or reviewing novel effectors designed to manipulate deformable materials infer a similar thinking pattern. Supporting statements are made by authors, including Koustoumpardis *et al.* [Koustoumpardis 04] who stated, “The investigation of the human performance must be the first step on the grippers design process” about fabric manipulation. Le *et al.* [Le 13] reinforce this perspective with the quote, “The careful study of how humans handle clothes can reveal the motion capabilities that are necessary and sufficient to efficiently perform virtually all likely manipulation tasks.” However, current robotic grippers produced for deformable manipulation do not generalise to various applications. Nor do they capture a complete set of manipulation skills needed to effectively grasp and sort clothing in a wide range of environments [Donaire 20, Sanchez 18]. One can argue that observing the capabilities of previous effectors with anthropomorphic definitions can inform design methodologies, control requirements and system scope. Beyond grasping mechanisms, research also looks to human strategies for more complex tasks in fabric manipulation such as folding, as seen in the video dataset present by Verleysen *et al.* [Verleysen 20].

Fabric, by nature, is a unique item to grasp. One aspect contributing to this description is the conformity of the object to the grasp action applied, enabling ambiguous wrist orientations and varied grasping strategies. In contrast, manipulating static objects involves constraining the wrist orientation and grasp poses to the target object [Feix 15, Roby-Brami 03]. Thus there are various descriptions of approaches to grasping clothing. One technique involves exploiting the environment while grasping fabric. A contributing factor to human beings’ generalised manipulation capabilities includes the skill of environmental exploitation, as discussed in Section 2.1.3. The taxonomy of Heinemann *et al.* [Heinemann 15] derives from this larger body of research by Eppner *et al.* [Eppner 15] and focuses on the hypothesis that humans exploit extrinsic contact to balance uncertainty when grasping. Several important conclusions were observed that apply to the manipulation of fabrics. First, all closing actions exploiting the environment during a grasp took on a similar form. Second, exploiting the environment can account for visual deficiencies. Finally, when one considers fabric manipulation, exploiting the environment is necessary for any generalised manipulator, as fabrics can present themselves in a flattened state requiring such a

skill. Grippers targeting fabric that consider this capability include the research of Koustoumpardis *et al.* [Koustoumpardis 14], Le *et al.* [Le 13], and Donaire *et al.* [Donaire 20].

2.1.5 Concluding Remarks

Human-oriented designs and approaches to grasping play a significant role in developing grippers targeting fabric handling and engineering broader grasping strategies. This section has discussed the skilful capabilities of humans regarding fabric manipulation, focusing mainly on grasps that leverage the environment to ameliorate uncertainty. Previous devices developed have reflected a theme of anthropomorphic-inspired grasping. Existing taxonomies for understanding human grasp pose and dexterous manipulation behaviour can assist researchers in developing human-inspired grasping strategies. However, previous research has not investigated robotic fabric grasping with these classification schemes. As a result, current gripper solutions remain limited and developing generalised solutions could usefully involve anthropomorphic classification schemes that identify novel gaps.

2.2 Dexterous Robotic Manipulation

2.2.1 Robotic Grippers

Robotic gripper design is a well-studied and evolving field. Bicchi *et al.* [Bicchi 00] present an early survey from the mechanical perspective, which discusses the literature on robotic manipulation throughout the latter years of the twentieth century. Discussed topics include grasping methods and the complex contact interactions between the hand and object. At the time of publication, the authors specifically mention how many manipulators within industrial applications focus on simple grasping behaviour and not dexterous manipulation. They also discuss enveloping grasps, meaning the grip in which humans use their fingers and palm to wrap around a target object to grasp and restrain said item. Generally, grasping involves the static equilibrium of mechanical systems where wrenches are exerted at contact points to maintain holds over objects. However, factors including friction at the contact points, control aspects of the manipulator utilised, and kinematic parameters will impact the system's grip on an object. When one considers grasping acts involving the environment or dexterous in-hand manipulation, the underlying dynamics are essential in modelling any robot hand. Bicchi *et al.* [Bicchi 00] also discuss contact compliance and how such an effect can be challenging to model.

Billard and Kragic [Billard 19] present a recent survey on robotic grasping discussing modern ap-

proaches and challenges in robotic manipulation. The authors outline existing obstacles in dexterous manipulation, describing how existing robots fail to demonstrate fine motor dexterity comparable to humans. In addition, robots face sensing challenges in dynamic environments that include occluded or unrecognisable objects. Data-driven approaches can encourage robust and adaptive behaviour. However, training on hardware can be expensive. Thus many researchers fall back on simulated environments. Billard and Kragic [Billard 19] then discuss two paths, learning manipulation from human observation and learning from realistic simulations. While both research paths have their benefits, each contains distinct drawbacks. For example, learning from observation is difficult to generalise, while existing simulators struggle to replicate friction, deformations, and other realistic physical properties. Thus while learning techniques can assist in understanding embedded dynamics within manipulation primitives, they are not a solution for all grasping challenges. They also come with development issues associated with machine learning techniques such as hyperparameter tuning.

Basic Structures

Grippers can be linkage-based architectures consisting of serial-chain or parallel devices. Simple serial-chain devices are sequential joint-link pairs that construct a robot body. Such mechanisms have established modelling procedures [Corke 17, Sciavicco 12]. Compared to serial-chain manipulators, parallel mechanisms are more complex with a closed-loop kinematic structure.

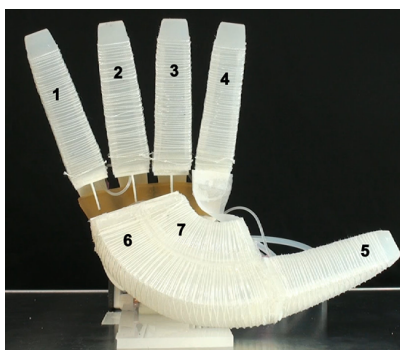
Many grippers are also underactuated, referring to systems with fewer degrees of actuation (DoA) when compared to the degrees of freedom (DoF)³. The benefits of underactuated systems include a reduced number of actuators within the system, which simplifies modelling procedures while remaining capable of enveloping grasps or shape adaption for robust grasping behaviour. Transmission mechanisms of underactuated systems can also include a wide range of technologies such as pneumatic, linkages and gearing. A survey by Babin and Gosselin [Babin 21] argues that underactuated systems can effectively comprise flexibility, simplicity and performance. Billard and Kragic [Billard 19] describe how designing highly biomimetic hands is a complex task due to the numerous DoF/DoA, the advanced tendon/bone structure of the human hand, and skin's sensitive haptic sensing capability. Many anthropomorphic designs currently limit the degrees of freedom/actuation and present a simplified serial-link structure. The human hand possesses features that are mechanically incompatible with traditional serial-link structures, and such differences must be accounted for when creating highly anthropomorphic designs. The high

³Some authors refer to the DoF of a TCP frame [Corke 17] or freedom within the mechanical structure [Ma 16].

level of actuation further increases control complexity.

Alternative Gripper Designs and Characteristics

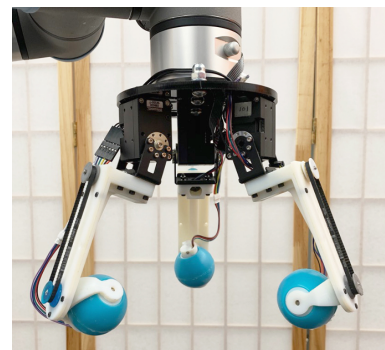
Babin and Gosselin [Babin 21] discuss alternative gripper designs, such as grippers with compliance or soft manipulators. Grippers can retain compliant mechanisms involving deformation within specific joints, eliminating mechanical issues such as backlash or clearance. However, compliant elements can make kinostatic modelling complex and require further study with Finite Element Method (FEM) analysis. For example, the prosthetic finger developed by Mutlu *et al.* [Mutlu 15] required said FEM analysis. Alternatively, soft grippers, manipulators characterised by soft materials, distribute grasping contact loads over object surfaces to reduce contact stresses. These devices include the RBO hand of Deimel and Brock [Deimel 16] or the Pisa/IIT Softhand introduced by Catalano *et al.* [Catalano 14] (Figure 2.3). A drawback of these soft manipulators is the complex modelling process which cannot utilise traditional modelling procedures for simple serial-chain or parallel linkage systems. Finally, the last type of gripper discussed by Babin and Gosselin [Babin 21] are active surface manipulators. These manipulators can alter the contact point of the target object with actuated contact surfaces such as belts or rotating surfaces which can provide physical transformations. Alternatively, one can observe mechanisms that utilise the effect of electrostatic adhesion. Several grippers implement active surface technologies. For example, the Roller Grasper by Yuan *et al.* [Yuan 20] was a three-fingered manipulator that could grasp and re-orient objects with actuated roller fingertips, see Figure 2.3(c). Regarding cloth manipulation, Abe *et al.* [Abe 20] and Yamakazi *et al.* [Yamazaki 21] implemented a robotic manipulator with roller fingertips to robustly grasp flattened fabric. The fingertips rotate, moving the contact point between the fingertip appendage and the target garment to improve grasping success. Alternatively, Donaire *et al.* [Donaire 20]



(a) [Deimel 16]



(b) [Catalano 14]



(c) [Yuan 20]

Figure 2.3: Examples of alternative gripper designs.

present a manipulator targeting fabric that retracts a surface with an increased friction factor to improve grip strength.

There are also various actuation and transmission mechanisms involved in manipulator development. Standard actuators include electrical, pneumatic or hydraulic devices. Electrical actuators are the most common modality due to their commercial availability and in-built control schemes. An example of this availability is the line of intelligent actuators such as the XM-430-350-R actuator [Inc 21], available from the company Dynamixel. Many manipulators also utilise cable or tendon-driven transmissions, resulting in lightweight, compliant appendages such as the Yale Openhand Project [Ma 17]. However, Babin and Gosselin [Babin 21] note how designing such mechanisms requires more complex considerations, such as retaining tension in the cables.

Sensors

Sensors are common within robotic grasping systems, informing actions such as dexterous manipulation or classifying held objects. More broadly, sensors provide another stream of data to inform a manipulator about the current state of the environment. Possible use cases include estimating a wrench exerted at a manipulator's tool-centre point (TCP), and detecting physical characteristics of a held object or slippage. Alternatively, sensors can take the form of simple joint encoders informing the current configuration of a robotic system. A recent comprehensive review of tactile sensing within the context of robotic manipulation by Yamaguchi and Atkeson [Yamaguchi 19] highlights that sensors applied to robotic grippers are a separate and broad field of study. Haptic exploration is frequently utilised in robotic fabric handling to deduce clothing features via tactile feedback from sensing modules. For example, Drigalski *et al.* [Von Drigalski 17b, Von Drigalski 17a] applied two triaxial force sensors at each fingertip to measure grasp force and classify materials. The classification process occurred by performing a linear 'rubbing' motion between the fingertips. As a result, it classified a wide range of household deformable materials, including fabric, belts, trash bags and pencil cases. Interestingly, Drigalski *et al.* [Von Drigalski 17a] demonstrate that features highlighting the friction co-efficient between objects produced a more accurate classifier of held objects within their collected dataset.

Similarly, the CloPeMa gripper of Le *et al.* [Le 13] also presented a two-fingered manipulator that performed a linear 'rubbing' motion between the fingertips for material classification. However, this end-effector used a multi-modal tactile sensor [Denei 17], made up of a capacitive pressure sensor array, microphone and ambient light sensor. The microphone and pressure array interpret data captured

during the rubbing motion performed by the end effector, and the ambient light sensor validates grasping success. Finally, specific haptic sensors applied to fabric can provide visual information. Yuan *et al.* [Yuan 18] and Luo *et al.* [Luo 18] use the Gelsight sensor with machine learning approaches to perform fabric classification. The Gelsight sensor contains a soft elastomer with an embedded camera to capture deformations. Such an approach captures the low-level geometry at a high resolution to classify held garments.

The examples cited above used tactile sensors for classification or grasp success, although sensing extends beyond fabric manipulation within custom gripper development. For example, Kaboli *et al.* [Kaboli 16] grasped a range of quasi-deformable materials with triaxial force sensors, targeting objects such as plastic cups or cardboard rolls. Their system estimates the friction coefficient between lifted objects and the gripper by opening the fingers until the held item begins to slip. This information modulates the grasp force. The sensor of the CloPeMa manipulator observed slippage detection through the microphone component as detailed by Denei *et al.* [Denei 17].

Slip detection has been a main area of research; Yamaguchi and Atkeson [Yamaguchi 19] describe slip detection mechanisms from the modalities of force, the centre of pressure and vibrations. They also describe the typical slip-detect-react process in which the grasp force is adjusted depending on detected slip parameters. A recent development in triaxial force sensors is the PapillArray sensor presented by Khamis *et al.* [Khamis 18, Khamis 19]. This sensor uses a silicon body and pinhole camera to estimate forces with reflective data produced by deformations of the silicon body. Khamis *et al.* [Khamis 21] subsequently showed how an array of these sensors could dynamically inform the near-optimal (in their case, near-minimal) grasp force for a range of held objects of varying weights and other physical properties.

Billard and Kragic [Billard 19] state how the advanced sensing capabilities of human skin also provide valuable information about forces, temperature and stretch. Traditional manipulators currently apply force sensors to the fingertips, while sensing in locations such as knuckles or phalanges may be crucial for more advanced manipulation. Currently, the impressive haptic sensing capabilities humans rely upon for dexterous manipulation is difficult to replicate and translate into robotic systems. However, research such as the human-robot interfacing framework suggested by Seminara *et al.* [Seminara 23] is investigating this topic. The literature indicates that sensors are a key aspect of robotic manipulation and have been utilised extensively within grippers targeting fabric applications. They also presents a modality that can assist in learned dexterous behaviour. Therefore, integration of haptics may be essential in grippers that target this dexterous manipulation problem, depending on the target applications.

Anthropomorphism

Many manipulators, both within the context of deformable manipulation and robotic grippers in general, consider the human hand and its dexterous skills the ‘holy grail’ of skilful manipulation [Huang 21]. As described by Babin and Gosselin [Babin 21], anthropomorphism can refer to the size, manipulation primitives and morphological features (palm and fingers) of a robot manipulator and how such parameters compare to the human hand. As manipulators reference or observe anthropomorphic behaviour, terms for human features such as phalanges are present when referencing mechanical structures [Koustoumpardis 14, Salviatti 15]. Some authors explore the topic of anthropomorphic replication; Xu and Todorov [Xu 16] present a robot hand replicating the bone and ligament structure of the human hand while attempting to develop a manipulator with comparable kinematics and dynamics. However, such a design approach can be intensive, with specialised components created to replicate these human components. Cable-driven and underactuated mechanisms are usually present in highly anthropomorphic designs, such as the grippers of Xu and Todorov [Xu 16], or Gosselin *et al.* [Gosselin 08]. Billard and Kragic [Billard 19] argue for designs beyond human anthropomorphism, targeting other animals, for example, the lamprey-based effector of Ku *et al.* [Ku 20], or even replicating multiple components of the human hand for novel designs, e.g. using two or more thumbs.

Anthropomorphism can also qualify a manipulator’s capability. For example, a preliminary version of the GRASP taxonomy [Feix 09] evaluated the dexterous capabilities of the RBO hand [Deimel 16]. The evaluation occurred by replicating grasps from the taxonomy on the developed gripper. Alternatively, anthropomorphic behaviour can inspire specific designs, which is common in grippers addressing fabric manipulation. For example, the gripper of Koustoumpardis *et al.* [Koustoumpardis 14] developed a manipulator with two fingers, a thumb and a palm. An observed human manipulation primitive involving the thumb, index and middle fingers inspired the design. A similar approach saw Shibata *et al.* [Shibata 09] use anthropomorphic wiping and pinching motions to develop a grasping mechanism.

Other Considerations and Final Remarks

Manipulation research continues to explore a series of unaddressed parameters and challenges. The review of Babin and Gosselin [Babin 21] discusses such parameters noting that, *inter alia*, one can consider the intrinsic dexterity of robot manipulators, namely, the ability to manipulate objects only with features of the robotic gripper. Within fabric manipulation literature, intrinsic dexterity can refer to a ‘grasp gaiting’ act [Borràs 20] where one adjusts the held grasp point of the garment by allowing

the material to slide between the fingertips [Donaire 20]. Alternatively, Sahari *et al.* [Sahari 10] present a manipulator that performs an inch-worm-like motion to adjust the held position on the fabric's body. Intrinsic dexterity can also refer to haptic exploration behaviour as previously discussed. In comparison, extrinsic dexterous behaviour exploits components that are part of the environment or held objects to enhance manipulation. In the context of fabric this mainly applies to exploiting the environment to assist with grasping behaviour. There is little or no indication of learned environmental exploitation grasping using tactile sensing within deformable manipulation. However, extensive research has investigated using reinforcement learning in contact-rich manipulation tasks [Elguea-Aguinaco 23].

Implementing grippers on robots involves difficulties such as grasp point localisation, motion planning and manipulation. Grasp point localisation relies on optimal grasping algorithms and scenario limitations. As Babin and Gosselin [Babin 21] discuss, grasp indices can assess the quality of a localised grasp point. Algorithms that also consider object parameters and gripper shape can enhance planning. While motion planning with manipulators can be a complex task, a common way of addressing this complexity is decoupling a manipulator's actions to the gripper, as seen regularly in fabric manipulation. Alternatively, Le *et al.* [Le 13] presented an environmental exploitation grasping act where the arm and gripper's compliant elements work together simultaneously to grasp a garment.

Measuring the capability or effectiveness of manipulators is also challenging. A common approach is to measure grasp success [Abe 20, Marullo 20], with some authors adding further conditions, such as holding an object for a specific duration after grasping [Donaire 20, Choi 18]. Newbury *et al.* [Newbury 23] attribute the lack of consistent performance metrics to the diversity of hardware, scenarios and objects involved in grasping. They also suggest using time-based metrics, such as mean picks per hour, defined as the average number of successful grasps over an hour. Such an attribute may assist in evaluating a complete system's grasping capabilities.

Additionally, Billard and Kragic [Billard 19] detail how existing complex manipulation actions include in-hand dexterous manipulation, navigating cluttered environments, object alteration (cutting/tearing), extrinsic dexterity, complex tool interactions and advanced bi-manual manipulation. They also detail how robots today can execute repetitive grasping patterns in semi-structured environments. However, occlusions or transparent objects can make manipulation difficult, additionally, as requirements change, robots must adapt to new tasks or challenges. Thus multi-purpose grasping solutions require further exploration. However, these solutions are a multi-modal problem requiring an understanding of the physical properties of new objects for grasping. Robots can infer this understanding from both vision

and haptic sensing, but, this inference is still limited. Current constraints in robotic manipulation involve the development of grippers as dexterous as human hands, and limitations in cognitive processing to develop robust and adaptable grasping capabilities.

While one can observe various advances in robotic manipulation, grasping in unstructured environments remains limited. In industry, common manipulators such as parallel pinch or vacuum manipulators rarely interact with the environment and remain decoupled from their manipulator. Advancing manipulation could involve optimising designs to handle extrinsic dexterous behaviour. Babin and Gosselin [Babin 21] conclude their review by highlighting the importance of utilising compliant elements for environment collisions, as any custom-designed manipulator should seriously consider the underlying physical interaction. They also state that decoupled arm-gripper pairs can result in robust manipulation strategies and design.

2.2.2 Targeting Deformable Manipulation

Introduction

The core aspect of this research focuses on developing a robotic manipulator for fabric manipulation. Throughout the literature, various specialised robotic gripping devices have targeted deformable manipulation. These devices have been limited to specific tasks and could not target rigid objects [Donaire 20]. Manipulation objectives such as extracting garments from a pile [Le 13], folding laundry [Le 15a], exploring a garment's body for haptic information [Von Drigalski 17a] or grasping fabric in a flattened state [Koustoumpardis 14] have been observed.

To the author's knowledge, the earliest study into end-effector design targeting fabric manipulation was presented by Parker *et al.* [Parker 83], who studied mechanisms such as pin catching (inserting pins into the fabric and applying pressure), adhesion or vacuum technology. Further research in the twentieth century included Ono *et al.* [Ono 91], who created a more anthropomorphic design with a pincer device that inserted a fingertip between layers. However, these early research projects targeted limited manipulation strategies, looking at the problem of extracting a flat garment from a neat pile. A review in 2004 by Koustoumpardis *et al.* [Koustoumpardis 04] advocated for more anthropomorphic strategies to grasp materials in the diverse range of conditions that they present themselves.

Before discussing more anthropomorphic designs, it is noted that several research reports present grippers with technical augmentations to assist with grasping actions, examples include the Mag-Gripper [Marullo 20] (Figure 2.4(a)) and the DressGripper [Dragusanu 22], which both use electromagnets to as-

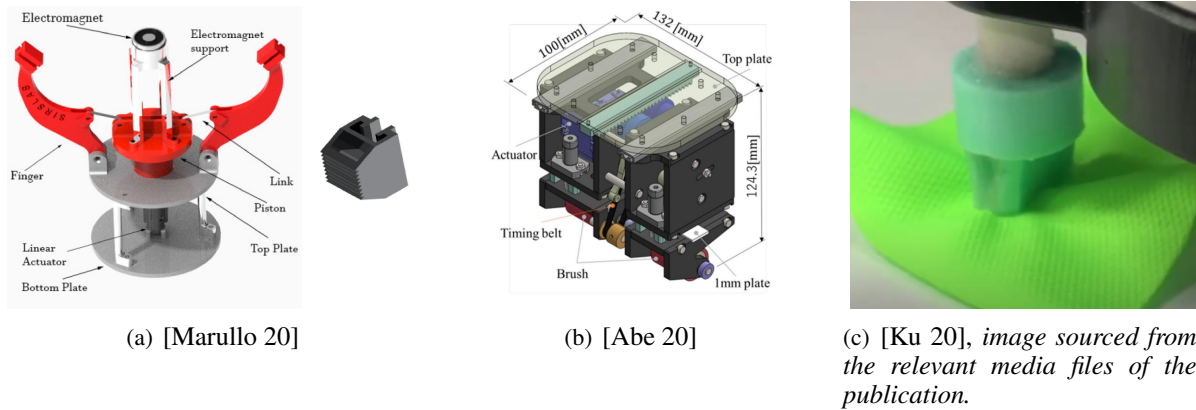


Figure 2.4: Examples of grippers with technical augmentations to assist with grasping behaviour.

sist in grasping fabric. Elsewhere, grippers with roller fingertips for manipulation purposes are reported. Sahari *et al.* [Sahari 10] presented a single effector with a roller fingertip, among others, for edge tracing tasks. Another effector design presented by Abe *et al.* [Abe 20] (Figure 2.4(b)) and Yamazaki *et al.* [Yamazaki 21] uses a roller fingertip to grasp flattened material from a surface or extract a single garment from a pile without disturbing the items beneath. A brush-like material surrounded the roller fingertip to ensure that the friction was appropriate for manipulating the range of fabrics within the project scope.

Another example is the gripper of Ku *et al.* [Ku 20] (Figure 2.4(c)), who present a soft gripper with embedded microneedles inspired by the mouth structure of the lamprey fish. The embedded microneedles assisted with a pinching act while the friction of the grasp and a vacuum mechanism created the holding force. Their results indicate that such a gripper could robustly grasp materials for delicate manipulations. Although these designs with technical augmentations are novel, they remain complex in fabrication and modelling procedures; a key benefit of anthropomorphic designs is the simplified development process.

Commercial manipulators applied to deformable manipulation

Some research has used commercial effectors to address deformable manipulation challenges, usually with respect to the tasks of intelligent cloth handling, grasp point localisation, state estimation, and motion planning. Maitin-Shepard *et al.* [Maitin-Shepard 10] used the PR2 robot with the default grippers for a towel folding task. Others, such as Kita *et al.* [Kita 11], utilised the HRP-2 robot in a clothing manipulation task. Further commercial effectors or robots utilised include the Baxter robot effectors [Li 15], the Barrett Hand [Balaguer 11] [Monsó 12], the Robotiq 3 Fingered Gripper [Ruan 18], the Shadow Hand manipulator [Twardon 15], the Nekonote robot arm [Moriya 18], a High Speed 3 Fingered Effector [Yamakawa 11] and the HIRO robot effector [Yuba 17]. As Borràs *et al.* [Borràs 20] note, many of these

research projects utilise what could be a simple two-fingered precision grasp for a wide range of tasks. Such an observation reinforces the view that a simple pinch or clamp grip can be effective for many fabric manipulation actions.

Grippers Designed For Fabric Manipulation

Several research projects have created custom grippers to address challenges in fabric manipulation. These devices can take the form of technically augmented devices, as previously discussed. However, many designs take inspiration from the human hand while addressing challenges within deformable manipulation. Such projects make consistent reference to anthropomorphic grasping behaviour, including the research of the CloPeMa project [Le 13], grippers developed by Ono *et al.* [Ono 91, Ono 01, Ono 07] and other authors [Koustoumpardis 04, Von Drigalski 17b, Shibata 12]. The development of grippers by these authors can be anthropomorphic in appearance, functionality, or behaviour.

For example, Koustoumpardis *et al.* [Koustoumpardis 14, Koustoumpardis 17], and Ono *et al.* [Ono 01, Ono 07] created three-fingered anthropomorphic effectors to mimic the grasping motions of a human thumb, index and middle fingers. These effectors could manipulate fabric against extrinsic contacts with EC grasping techniques. Alternatively, some grippers were based on two-fingered precision grasps. For example, the NAIST M2S OpenHand *et al.* [Von Drigalski 17b] was a manipulator that had a design based on a static thumb and index finger and was able to perform haptic exploration with embedded triaxial force sensors. Another instance of a two-fingered precision gripper was introduced by Le *et al.* [Le 13, Thuy-Hong-Loan Le 13] as part of the CloPeMa project. Rather than recreate a human hand, Le *et al.* [Le 13] built a simplistic gripper that could perform a subset of motions humans use to manipulate cloth. In addition, the effector had a multi-modal tactile sensor [Denei 17], which would validate grasp success and perform haptic exploration.

One crucial aspect of manipulating fabric is performing EC grasping behaviour. Several research projects developing custom manipulators target this task. A common approach was to pin the fabric to the table before dragging a finger structure along the table's surface to deform the fabric and produce a protrusion for grasping. Various authors have used this technique in deformable object manipulation [Ono 01, Ono 07, Murakami 04, Shibata 12]. An alternative approach applied by Le *et al.* [Le 13] and Donaire *et al.* [Donaire 20] offers another solution, in which a compliant thin fingernail-like appendage inserts itself beneath the fabric on the environment surface, followed by a clamp grasp.

While some grippers perform haptic exploration for tasks such as classifying held materials, grasp

gaiting is another function of interest, being the act of adjusting the grasp position on the fabric without relinquishing the grasp. Within fabric manipulation, such an action is usually performed by adjusting the grip strength and allowing the fabric to slide between the closed fingertips. Sahari *et al.* [Sahari 10] present three grippers that can traverse the bodies of garments using grasp gaiting. These grippers have alternative fingertip designs to perform this task, namely, a simple pincer gripper, a gripper with roller fingertips and a gripper designed to move in an ‘inchworm fashion’ across the fabric. Others have also demonstrated grasp gaiting, including Shibata *et al.* [Shibata 12], who present a dual sliding prismatic effector to manipulate a tablecloth, and Donaire *et al.* [Donaire 20], who adjust the friction between their gripper and the fabric to enable grasp gaiting by sliding.

Another aspect to consider when manipulating garments is the grasp force. While several projects detail grasping forces up to $5N$ [Von Drigalski 17b, Sahari 10], the gripper proposed by Le *et al.* [Le 13] can grasp with a strength of up to $40N$. Le *et al.* [Le 13] detail how grasping tangled fabric can require a maximum grasping force of $30N$ while Drigalski *et al.* [Von Drigalski 17b], and Sahari *et al.* [Sahari 10] do not consider this scenario. This observation broadly indicates that fabric can often require a reduced grasping force, but will require the capability to exert greater forces under specific scenarios. Donaire *et al.* [Donaire 20] do not detail the grasp force of their manipulator but evaluate the strength by holding a range of materials with differing weights.

To summarise, many authors have considered the unique skills required to manipulate fabric and have produced manipulators to address these challenges. The desired task scope, context, and applications inform the various designs. A significant trend amongst gripper design for garment handling looks to human manipulation for inspiration of design parameters. However, one can observe that few projects look in-depth into the literature on human manipulation, but rather, heuristically identify behaviours to replicate.

2.2.3 Serial-Chain Manipulators

The gripper described in Chapter 4 is a simple rigid body serial manipulator. Modelling of the kinematics and dynamics of such mechanisms are well documented [Corke 17, Sciavicco 12, Yoshikawa 90]. Therefore, modelling the gripper follows established procedures to define serial-chain systems. For example, Bellicoso *et al.* [Bellicoso 15] present a lightweight serial manipulator for drone-based applications. Such an approach uses homogeneous transformation matrices and the Denavit-Hartenberg (DH) parameters to model serial-chain manipulators. Using the defined DH parameters and kinematic variables,

one can utilise sequential transforms to determine the relative position of the robot's tool-centre point (TCP), or fingertip in this case, from the base of the manipulator. Such an approach is known as Forward Kinematic (FK) modelling. Furthermore, a closed-form set of equations or numerical algorithm can establish the joint positions from the TCP's position depending on the number of parameters present within the mechanical system, in an approach called Inverse Kinematics (IK). Such methods represent the positional relationship between the actuator positions and TCP's configuration.

The next step this standard modelling process uses Jacobian matrices to determine the relationship between the velocity of the joints and TCP. When modelling serial-link systems, Jacobian matrices use partial derivatives from actuator FK expressions alongside matrix multiplication with actuator speeds to accurately represent the magnitude and direction of the TCP's velocity. The Jacobian can also be modified to input desired TCP velocities to calculate required joint speeds. Finally, when modelling Kinostatics, the transposed Jacobian can also estimate wrenches exerted at the TCP and their relationship with the applied actuator forces and torques.

A dynamics analysis is the last component of any serial manipulator modelling process. This component concerns physical characteristics and the estimation of required torques depending on motion requirements. Such a calculation involves the dynamic parameters of gravity, inertia and the centre of mass (CoM) for the various links involved. The Lagrangian method presented by Sciavicco and Siciliano [Sciavicco 12] models this component. The Lagrangian function subtracts the potential energy from the kinetic energy available within a serial manipulator structure.

2.2.4 Prototyping novel manipulators with commercially available components

As Chapter 4 looks at constructing a novel manipulator with commercially available components, considerations for research that has developed low-cost robots are present. Several projects have produced legged robotics including Faigl and Čížek [Faigl 19], Trivun *et al.* [Trivun 17] and Bjelonic *et al.* [Bjelonic 18]. These projects use actuators produced by the company Dynamixel. These rotational actuators are available for hobby projects and more advanced research endeavours. These motors generally use PID loops with a magnetic encoder for position and velocity control. Such devices provide an immediate actuator with robust control for prototyping robotic systems.

More advanced control mechanisms, including torque and *Current-Based Position* control, are available in different models of actuators from Dynamixel. *Current-Based Position* control is recommended explicitly for grasping applications as it provides a convenient impedance controller, useful for system

motion and grasp-force modulation features. One can also observe these actuators in research manipulators, including the manipulator targeting cloth manipulation of Donaire *et al.* [Donaire 20]. Other manipulators that also use Dynamixel servo motors include the **D’claw** presented by Ahn *et al.* [Ahn 20] or the universal gripper proposed by Choi *et al.* [Choi 17].

Additionally, the proposed manipulator created in this thesis also includes a triaxial force sensor provided by the company Contactile⁴. This company produces the commercial versions of the sensors proposed by Khamis *et al.* [Khamis 18, Khamis 19, Khamis 21].

2.3 Reinforcement Learning

To imbue the novel manipulator in this research with appropriate learned dexterous behaviour, one can look to Reinforcement Learning (RL) as a possible solution. RL, as described by Sutton and Barto [Sutton 18], is a research field concerning the problem of teaching an agent how to interact within an environment. These agents include characters in a physical simulation, robots, mechanical components, or vehicles. Essentially, the field involves observing the world and teaching the agent to perform optimal actions through rewards. RL can apply to problems that require intelligent decision-making and responsive behaviour in dynamic conditions. These factors have made RL a popular approach to intelligent manipulation [Han 23] and contact-rich tasks [Elguea-Aguinaco 23]. The research reported in Chapter 5 focuses on learning grasping motions that leverage the environment using RL. Therefore, this section of the literature review discusses how RL has previously produced robust autonomous skills alongside approaches to achieve these results. Topics discussed include challenges in RL, training methods, algorithms, and implementation details.

2.3.1 Introduction

Before discussing algorithms applicable to continuous control in robotics, this section briefly outlines key terms and foundational concepts. A policy in RL terms is a type of controller (π_ϕ) that informs an agent how to interact within an environment by using the state (s) to derive an action (a). Policies can either be deterministic, $\pi_\phi(s) = a$, or stochastic, $\pi_\phi(a|s) = \mathbb{P}_{\pi_\phi}[A = a|S = s]$. Deterministic policies derive a single action from a state, while stochastic policies determine an action based on a learned probability distribution. While both stochastic and deterministic policies can apply to continuous control

⁴Company information at the time of thesis writing is at the link <https://contactile.com/>.

in robotics, stochastic policies can stabilise training [Haarnoja 18b]. However, Haarnoja and colleagues also saw a performance improvement when switching to a deterministic policy for deployment.

Policies learn to take optimal actions by the maximisation of a reward signal, or maximising the return, denoted as G_t . One defines the return as the sum of rewards across the trajectory. However the discount factor γ , a foundational component of reinforcement learning [Sutton 18], penalises rewards beyond the current time-step. Mathematically, the return can be represented as $G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$. The policy receives a reward value, r , upon every action taken. When defining interaction between an agent and the environment, the representation involves a sequence of actions and obtained rewards in time $t = 1, 2, \dots, T$, with T representing the time at the terminal (completion) state. Throughout the trajectory, the agent receives rewards by interacting with the world, therefore learning the optimal policy that maximises G_t .

Usually, one denotes the state, action and reward at time t as s_t , a_t and r_t . The policy obtains a reward as it reaches the following state. Thus a whole trajectory, including the terminal state s_T , can be expressed as $s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, \dots, s_T$. This sequence of state, action, reward and state forms a *transition step*. While in the state s , the policy takes action a to reach the next step s' and achieve a reward r , resulting in the *transition step* tuple (s, a, s', r) . A terminal state can occur when a constraint violation occurs, or the trajectory completes a goal ending the task. Tasks following this trajectory type are episodic, meaning a clear beginning and end to a trajectory exist (known as an episode). Individual episodes will also not affect others. Alternatively, a trajectory can be continuous, meaning a task or learned skill does not break down into episodes but instead continues indefinitely [Sutton 18].

There are two classes of RL learning. Algorithms can either be *model-based* or *model-free*. A model in RL acts as a descriptor of the environment. Model-based RL algorithms assume a perfect world model. Therefore, dynamic programming (DP) techniques or function approximators will learn the environment's dynamics model, which is then applied to a control or planning algorithm to maximise the reward. Model-free RL targets problems where the underlying dynamics of a system are unknown or where a set of limited parameters (e.g., sensors on a robot) estimate the environment dynamics. Thus, learning does not depend on a perfect representation of the world. Furthermore, one can classify model-free approaches as either on-policy or off-policy approaches. On-policy methods use samples from the target policy (i.e., the actively learning policy) to train the algorithm. In contrast, off-policy algorithms refer to learning that trains on collected samples or distributions produced from a behaviour policy rather than the target policy. Sutton and Barto [Sutton 18] provide further information surrounding the model

within RL.

One can formulate most RL problems using Markov Decision Processes (MDPs). Any state within an MDP should be Markovian (retaining the Markov property), meaning that the future can be estimated with the current state, regardless of past actions. Essentially, all information within the current situation can estimate the future reward. An MDP is represented mathematically as the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$. The symbols represent a set of states, actions, the transition probability function, a reward function, and a discount factor respectively. If a complete understanding of the model or environment is available, solving an MDP becomes a planning problem of directly applying Bellman equations using dynamic programming algorithms. However, within learned robotics, the model dynamics are often unknown, thus requiring SOA RL approaches. Modern algorithms applicable to continuous, complex robotics build upon foundational RL concepts, including temporal difference (TD) learning, Q-Learning, the policy gradient (PG) theorem and actor-critic (AC) frameworks [Sutton 18].

2.3.2 Algorithms

A broad spectrum of modern RL algorithms can apply to solving challenging robotic tasks, including PPO (Proximal Policy Optimisation [Schulman 17]), TRPO (Trust Region Policy Optimisation [Schulman 15]), QT-Opt [Kalashnikov 18], policy gradient methods [Peters 06] or DAPG (Demo Augmented Policy Gradient [Rajeswaran 17]). However, Ibaraz *et al.* [Ibarz 21] detail another popular approach to learning robust skills with a high-sample efficiency using off-policy AC algorithms such as Twin Delayed DDPG (TD3, [Fujimoto 18]) and Soft-Actor-Critic (SAC [Haarnoja 18c]). These algorithms fall into the model-free off-policy classification, as they utilise the experience replay mechanism. They also can effectively apply to problems without a clear model structure or unclear environment dynamics, making them suited to learning intelligent control for tasks too complex for a handcrafted approach. These off-policy approaches have seen success in learning tasks such as locomotion [Zhu 22] and manipulation [Ahn 20].

Initially proposed in 2018, SAC and TD3 remain effective across various robotic applications as Section 2.3.3 outlines. However, each algorithm remains unique regarding the theoretical framework that constructs the learning object alongside minor implementation details. TD3 by Fujimoto *et al.* [Fujimoto 18] presents an expansion of the Deep Deterministic Policy Gradient (DDPG) algorithm [Lillicrap 15], which presented a breakthrough of an AC policy gradient algorithm applied to continuous action space problems. This algorithm targeted a range of tasks using a deterministic policy and demonstrated

significant improvement over previously existing methods such as Deep Q-Network (DQN, [Mnih 13]). DDPG contains four neural networks in the algorithm structure, two acting as actor and critic function approximators alongside two more target variants of these networks.

DDPG combines elements of DQN and the Deterministic Policy Gradient (DPG) algorithm [Silver 14], with the critic networks updating similarly to DQN, in which a Q-network learns with gradient descent and a loss function inspired by the Bellman equation. The actor networks update in an analogous manner to DPG using gradient ascent methods with a loss function based on the total expected return. DDPG presents several mechanisms to stabilise the learning process, including minibatch gradient updates, in which the learning process samples a user-defined number of transitions from the replay buffer and takes the mean of the sum of gradients to estimate the objective gradient. To improve exploration, DDPG also applies noise to taken actions. DDPG also utilised batch normalisation, initially presented by Ioffe and Szegedy [Ioffe 15], to normalise physical parameter inputs. This method particularly applies to robotics, whose state and action spaces contain velocity or position features. Finally, DDPG implements a ‘soft’ target update. Unlike DQN, which freezes the target network for a number of steps, DDPG smooths the updating action of target networks by slowly tracking the weights of their non-target counterparts. For convenience, this thesis will refer to this technique as a ‘soft update’ moving forward.

However, DDPG suffers from several limitations, which TD3 [Fujimoto 18] addresses. Simply put, Q-Learning can generally experience overestimation bias of the value function [Thrun 93]. Furthermore, TD learning uses a bootstrapping technique to update the value function. Thus, an accumulated error will occur when estimating the value function if each update retains a small margin of error. Fujimoto *et al.* [Fujimoto 18] discuss how these problems persist in DDPG and suggests three significant alterations to the algorithm resulting in substantial improvements.

The first change is the integration of dual critic networks to evaluate a single policy. Then, while training the policy, TD3 applies a technique called clipped double Q-learning which takes the minimum estimation between the two critics, avoiding overestimation. The second change reduces the policy and target networks’ update frequency compared to the critic networks, which reduces variance. In this case, the policy, target policy and target critic networks are updated at a lower rate when compared to the critic networks. The third and final alteration of TD3 is the target policy smoothing regularisation method. TD3 utilises a deterministic policy, so it remains possible that the policy will over-fit to narrow peaks in the value function. To combat this, TD3 applies a clipped noise term to the actions taken by the target policy network, which then updates the target critic networks. These proposed changes significantly improved

the learning capability of AC frameworks and resulted in a SOA method for learning continuous control tasks.

At the same time TD3 was published, the SAC algorithm [Haarnoja 18b] was also presented. SAC is an actor-critic algorithm that utilises the maximum entropy framework (MEF), which attempts to succeed at a task while acting as randomly as possible. SAC achieves this goal by using an objective function that incorporates an entropy measure term which encourages the policy to explore. A user-defined hyperparameter (known as the *temperature*) determines the impact of the entropy measure on the learning objective. Haarnoja *et al.* [Haarnoja 17] previously used the MEF framework while developing Soft Q-Learning. However, SAC presented the first off-policy MEF AC algorithm making such an approach unique. SAC has two versions, both from Haarnoja *et al.*; the first, [Haarnoja 18b] presented the initial algorithm, and the subsequent article, [Haarnoja 18c] discusses several enhancements. For brevity, this section only observes the improved version discussed in the more recent article [Haarnoja 18c].

Unlike TD3, SAC uses a stochastic policy. Thus, SAC gives an equal probability to optimal actions in situations where multiple options are available. SAC consists of components known as a soft policy (the actor) alongside a soft Q-function (the critic). These components use loss functions that consider the MEF. For each policy update step, SAC directs the policy towards the exponential of the soft Q-function by minimising the Kullback-Leibler (KL) divergence [Haarnoja 18b]. The soft Q-function updates via a loss function that minimises the soft Bellman Residual, which is a variation of the Bellman Residual that incorporates the entropy regularisation term introduced by the MEF. The refined version of SAC also takes inspiration from TD3 and uses two independent soft-Q-functions. The minimum value from these two soft-Q-functions apply when updating the policy in an analogous manner to clipped double Q-learning. The updating of the target soft-Q-functions also make use of ‘soft updates’ to stabilise training.

Finally, the version of SAC presented by Haarnoja *et al.* [Haarnoja 18c] noted a limitation in the initial SAC algorithm in which training success was dependant on selecting the correct entropy temperature hyperparameter for different tasks. In order to mitigate this issue, the revised SAC algorithm formulates a constrained optimisation problem, in which the policy should satisfy a minimum entropy constraint while maximising the expected return. Such an approach tunes the impact of the entropy measure on the learning objective by adjusting the temperature hyperparameter. This method requires an additional hyperparameter representing a target minimum entropy threshold.

While TD3 and SAC demonstrate success in learned robotic skills, more advanced algorithms have recently been released, demonstrating a greater ability to learn various complex tasks and at a greater

level of efficiency. A recent example is the Temporal Difference learning for Model Predictive Control algorithm (TD-MPC) by [Hansen 22]. TD-MPC is a framework that learns a task-oriented latent dynamics (TOLD) model alongside a terminal value function using TD learning. Broadly, this algorithm learns the underlying dynamics of the model that are predictive of reward for shorter trajectory segments with a value function that guides long-term planning. The resulting algorithm solved complex tasks in the RL domain, including the Dog environment from the Deepmind control suite [Tassa 18], a 38-dimensional continuous action space task for which SAC failed to converge to a solution.

This approach of RL with model predictive control (MPC) can also be considered a combination of model-free and model-based learning. Others have also explored such a concept, including the proposed LOOP algorithm of Sikchi *et al.* [Sikchi 22], who similarly extend SAC with a learned model and planning (with an H-step look-ahead method). Alternatively, one can observe data-augmentation methods such as DrQ [Yarats 20, Yarats 21] who augment image-based RL tasks with SAC and DDPG, demonstrating improvements with image-based RL. Another variation on SAC includes meta-learning adjustments [Wang 20b], who propose Meta-SAC, an expansion on the second version of SAC [Haarnoja 18c] which automatically tunes entropy impact using a meta-learning approach rather than the constrained optimisation approach presented by Haarnoja *et al.* [Haarnoja 18c].

While such approaches can significantly improve training, sample efficiency and stability, the previously discussed TD3 and SAC are established benchmarks and have been repeatedly applied successfully to learn robotic skills such as manipulation [Ahn 20, Gupta 21, Schoettler 20] or locomotion [Ahn 20, Zhu 22, Haarnoja 18a]. Many authors initially discuss how these baseline AC off-policy algorithms can learn specific skills with enough data and the appropriate hyperparameters. Additionally, the actual task of EC grasping, which the proposed manipulator of Chapter 4 learns, remains a low action-dimension task with a relatively specific desired trajectory. Therefore, SAC and TD3 remain stable, appropriate solutions to the desired learning outcomes of Chapter 5.

2.3.3 Overview of Learned Robotic Skills

Robots and Learning Skills

Ibarz *et al.* [Ibarz 21] discuss key lessons and considerations about training robotic systems. While the authors do not present a comprehensive overview of RL applied to robotics, they provide an in-depth discussion of previous challenges encountered by an experienced research group when developing learned robotic systems. Usually, the goal in any RL problem is optimising a controller (referred to as

a policy in an RL context) to achieve the maximum possible reward while completing a goal action or operating in a continuous space. However, learning with robots presents several initial barriers, including cost, safety, hardware considerations, and a learning setup where sensors infer the world state.

Robots present several elements in simulation and hardware which can complicate the learning process. Robot actuators will use control methods such as traditional PD controllers, PID gains, torque control, and impedance control, to move the system to desired configurations. Depending on the application, systems usually implement a specific low-level control algorithm for individual components while performing deep RL training procedures. An example is the ROBEL (Robotics Benchmarks for Learning) benchmarking suite [Ahn 20], which uses position and velocity PID loop control to learn the tasks of rotating a valve and locomotion. For many tasks, position and velocity control will suffice for various applications. However, tasks that are contact-rich require control that can handle external perturbations. Varin *et al.* [Varin 19] explored several contact heavy manipulation tasks, i.e. hammering, peg inserting and object pushing, against differing low-level control mechanisms such as proportional derivative (PD), torque, inverse dynamics (ID) and impedance controllers. Across all tasks, their results indicate that low-level impedance controllers paired best with deep learning algorithms PPO and SAC. In addition, robotic hardware is known to degrade over time and policies deployed on hardware may see a decrease in performance. Lifelong learning or online adjustment methods can assist with such phenomena [Ibarz 21].

Ibarz *et al.* [Ibarz 21] further discuss case studies and outstanding areas requiring investigation in robotic RL. The case studies discussed learning manipulation and locomotion, including how off-policy methods are more sample efficient than their on-policy counterparts, requiring far fewer training steps throughout learning. One can expect such an effect as off-policy algorithms can learn from the complete data collected across the training process. In contrast, on-policy methods are usually more stable throughout learning than off-policy approaches. Model-based learning techniques are even more sample efficient than model-free techniques. However, model-based techniques are less applicable to learned autonomous skills in a continuous domain due to the perfect mathematical representation of the environment required.

TD3 and SAC have been widely adopted in robotic learning, with SAC being one of the algorithms used by the ROBEL benchmarking suite [Ahn 20]. This suite of algorithms and tasks learned skills in several low-cost robots, the D’Claw manipulator and the D’Kitty quadruped, actuated by Dynamixel servomotors. As described above, the D’Claw learned how to turn a valve to an appropriate angle, and

the D’Kitty learned locomotion alongside acts of standing up and reorientation of the body. These tasks were learned by reading kinematic variables such as joint positions and velocities alongside relative error to the task objective.

Due to its off-policy implementation, SAC robustly learned policies more efficiently than the other algorithms on the ROBEL benchmark, highlighting that such approaches can be appropriate for unsupervised learned robotic skills with unknown dynamics and lower-level controllers. Actions given by these policies in this task were simply the desired joint positions for the robotic platforms. Alternatively, Varin *et al.* [Varin 19] pursued their analysis of action spaces using a range of kinematic and dynamic commands, including direct joint torques, desired joint positions/velocities and desired TCP positions/velocities. In similar research, Beltran-Hernandez *et al.* [Beltran-Hernandez 20] used SAC to learn force control with a UR3e robot arm that generated trajectories and tuned optimal position control parameters.

Using kinematic or dynamic joint states is common across research in learning advanced robotic skills. Peng *et al.* [Peng 20] use a combination of previous joint poses and actions taken in the state vector. In their case, actions were defined as target rotations for a quadruped’s lower-level joint PD controllers while learning to replicate animal gaits on a robotic platform with PPO. Alternatively, Zhu *et al.* [Zhu 22] learn locomotion with TD3 alongside trajectory optimisation techniques for optimal energy use policies. The action space was control commands to actuator positions using position control, and the state-space contained parameters from the original ‘Ant’ PyBullet model, including poses, velocities, Cartesian coordinates and external forces. Alternatively, Wang *et al.* [Wang 20a] present a learning problem surrounding mobile manipulation. This project aimed to grasp an object within the environment using a Husky rover with two UR5 robots. The state was a vector informed by Cartesian coordinate positions of the object to grasp (inferred from vision modules), the gripper, and robot base frames. Although PPO resulted in the most stable training procedure and policy, TD3 was the quickest algorithm to converge to a solution but suffered from a wide variance in evaluation procedures, highlighting the improved stability on-policy approaches usually see.

Variations exist in robotic learning action spaces, whether using joint positions or Cartesian coordinates calculated with FK algorithms. Ganapathi *et al.* [Ganapathi 22] discuss how robotic deep learning projects usually determine whether action spaces should use calculated Cartesian coordinates or joint spaces using the task context and heuristics. They argue that both details are valuable and embed the information into deep learning tasks with a method they call Implicit Kinematic Policies (IKP), showing

improvement or equal performance under testing. RL goes beyond learning specific skills with mathematical robot information inferred from sensors. Ibarz *et al.* [Ibarz 21] mention how TD3 and SAC have produced significant results in learning to manipulate using vision. For example, Matas *et al.* [Matas 18] used RGB images and proprioceptive information to learn several fabric manipulation-based tasks using DDPG with enhancements (including elements from TD3). Other research projects have also seen success with robotic RL using visual data streams [Kalashnikov 18, Singh 19, Zhu 20]. As discussed throughout this section, SAC and TD3 have successfully learned various robotic skills through vision-based intelligence and kinematic system response. While some argue whether TD3 or SAC can produce more robust behaviour, the broad consensus concludes that either algorithm can produce similar results with optimal hyperparameters. In their research, Yang and Nguyen [Yang 21], and Chan *et al.* [Chan 19] note a similar performance between both algorithms.

Robotic Simulations For Deep RL

Ibarz *et al.* [Ibarz 21] also discuss simulations in the context of deep RL. Benefits include improved sample efficiency and robot safety, as robots may produce erratic and noisy behaviour upon beginning a training process. Some approaches learn initially in simulation before fine-tuning behaviour on a real-world platform [Tan 18, Peng 18, Peng 20, Rao 20]. However, training in simulation before moving to hardware meets a challenge commonly known as the *reality gap*, a degradation in performance when transferring learned policies from simulation to the real world. The reality gap can appear for various reasons, including the nature of robotic hardware, which is noisy, and factors such as latency and incomplete system observability impact the deployment of learned policies. The approach of fine-tuning trained policies on hardware is one method to overcome this degradation but requires a two-step training process while also handling safety concerns when the transfer to hardware occurs.

Collins *et al.* [Collins 19] presented research that looked at the behaviour of simulations and attempted to quantify the reality gap. They discuss and evaluate platforms, including MuJuCo, PyBullet and V-Rep (CoppeliaSim). While basic kinematics and control could perform well, simulations demonstrated difficulty replicating complex collision scenarios from the real world. The broader literature on optimal robotic simulations also reflects that no simulation is uniformly better than its peers [Chung 16, Erez 15]. Collins *et al.* [Collins 19] detail how the PyBullet simulation (Coumans and Bai [Coumans 16]) returned the lowest error in two of three evaluation scenarios.

Other researchers, such as Körber *et al.* [Körber 21], detail how PyBullet and MuJuCo target Deep

RL applications and highlight that PyBullet is more accessible to researchers within an academic context. They also note that PyBullet was stable at simulation speeds above or equal to $7ms$. In addition, Collins *et al.* [Collins 21] addressed robotic simulations in a broader context evaluating both robotic tasks and environment settings. They discuss how simulators such as MuJoCo, Gazebo, V-REP, and SimGrasp can apply to manipulation. While each has its benefits, including kinematics estimation, dynamics integration, and ROS compatibility, the open-source PyBullet has specific benefits for deep learning applications, including accessible benchmarks, RL environments, and compatibility with robotics tools.

Thus for basic manipulation primitives, PyBullet fulfils the criteria for robotic deep RL research, especially when considering rigid body manipulators with collision handling, which is particularly applicable in the learning process outlined in Chapter 5. Additionally, the simple python API makes interfacing to ROS a relatively simple coding task [Collins 19]. Additionally, PyBullet can act as an appropriate platform for rapidly deploying and evaluating simulation environments. While Gazebo can provide a simulation with a direct interface to ROS, PyBullet also provides examples of RL applications, and many users have noted the simple setup and accessibility of the platform, e.g. Körber *et al.* [Körber 21].

Regarding PyBullet, many recent RL projects have successfully used this simulator to train locomotion and manipulation policies and transfer them to the real world. For example, Peng *et al.* [Peng 20] used PyBullet to learn locomotion and transfer policies to the Laikago robot platform. Tan *et al.* [Tan 18] also used PyBullet as their simulator for learned locomotion, which was then directly applied to the hardware platform. Many additional examples have learned policies through PyBullet before being tuned or deployed in the real-world [Kaspar 20, Church 22, Matas 18]. As a point of interest, Matas *et al.* [Matas 18] target transferring fabric manipulation tasks from PyBullet into the real world using elements of TD3 and demonstrations.

Methods to Address the Reality Gap

One of the significant sources of error contributing to the reality gap is the nature of hardware communication, which is asynchronous. Thus, the active policy will dictate actions while new state information is available. Additionally, the data from sensors may not accurately represent the current state but an estimated configuration affected by latency or noise, resulting in many RL algorithms experiencing difficulty upon deployment to hardware. Consequently, asynchronous control and sensor-based estimations result in non-Markovian systems, which is problematic to the MDP formulation of these problems. The Markov principle expects actions to lead to new states that encode all the necessary information to exe-

cute subsequent behaviour, however, these non-Markovian aspects make formalising real-world robotic problems with an MDP complex.

The formal definition of an MDP states that all information required to decide an action exists solely in the current state. When taking external information such as previous actions or observations outside the current state to inform decisions, a solution becomes non-Markovian. Ibarz *et al.* [Ibarz 21] reference how hardware consistently presents challenges within learned robotics applied in the world, citing reasons of partial observability, safety boundaries and unknown dynamics alongside the previously mentioned issues within real-world communication. In addition, they outline how specific methods are required to overcome the non-Markovian aspects of hardware. One solution is incorporating real-world phenomena such as latency into simulations. Alternatively, they suggest adding recurrence to model-free policies by incorporating a window of observations or appending previous action commands to the state. Treating RL problems with recurrent neural network (RNN) frameworks formulates the learning problem as a Partially Observable Markov Decision Process (POMDP). Formalising a problem as a POMDP can assist with several issues in learning, including the reality gap, a limited estimation of the system model or complex behaviour dynamics. Xiang and Foo [Xiang 21] present a review of RL within the context of MDP and POMDP problems.

Meng *et al.* [Meng 21] reinforce this perspective, describing how partially observable conditions frequently appear in hardware-based systems. They discuss how POMDPs formally differ from the standard MDP by altering the traditional mechanics observed. For example, one can represent a traditional MDP as the tuple $\langle S, A, P, R \rangle$, respectively representing the state space, action space, transition probability and reward function. At the same time, a POMDP expands this representation in the form of $\langle S, A, P, R, O, \Omega \rangle$ where the value of O represents an observation space and Ω is an observation model. The critical difference in these representations is that the agent cannot receive the model state of the system. Thus an observation is given upon a transition with Ω acting as a transition function.

To address POMDPs, neural network architectures that take in historical data surrounding robotic behaviour sequences can create robust policies. Meng *et al.* [Meng 21] present an example of such an approach which adapted the TD3 algorithm to a novel approach referred to as LSTM-TD3, which integrated Long Short-Term Memory (LSTM), an artificial RNN architecture. They used this framework to consider the previous states and actions alongside the current observation. They also attempted variations in function inputs, including past action/observation concatenations. Previous RL approaches inspired this research and demonstrated that recurrent architectures are well suited to POMDPs. Another

example is Song *et al.* [Song 18], who utilise the Recurrent deterministic policy gradient algorithm in an RL walker challenge. They found that treating the problem as POMDP with a recurrent architecture performed significantly better than an MDP formulation.

Meng *et al.* [Meng 21] show how their proposed algorithm, LSTM-TD3, navigated various environments from the gym library while deliberately violating the Markov assumption by modifying the state vector with various adjustments. These adjustments included adding noise to the data, dropping data packets and removing modalities such as sensors or physical information such as velocity. For tasks such as the ant, walker or hopper virtual environments, TD3-LSTM consistently performed the best. For the half-cheetah environment, regular TD3 performed optimally under the MDP condition, while TD3 augmented with previous observations in the state space or LSTM layers achieved the best results under POMDP conditions. Both pendulum environments from the gym library displayed similar results to the output from the half-cheetah environment. These results by Meng *et al.* [Meng 21], together with the broader literature, indicate that for imperfect information coming in, such as real-world robotics, RL could benefit from treating these projects as a POMDP. While unconfirmed, such a technique may also assist in overcoming the reality gap.

Within the broader literature, these approaches are applied successfully for various tasks. For example, the ROBEL benchmark of Ahn *et al.* [Ahn 20] observed SOA RL algorithms for low-cost robotics in the real world. The tasks include moving to a pose, in which the reward was a combination within the kinematic parameters of joint speed and position. The rewards were given each time step and estimated by relative error to the goal configuration. For all tasks on these low-cost robots, the observation space includes the previous actions, moving the problem formulation closer to a POMDP. Peng *et al.* [Peng 20] use sequences of past actions and states in their state vector to kinematically replicate animal gaits on the Laikago quadruped. Usually, including the last action taken in the state vector would violate the Markov property. However, this line is blurred in robotics as the last actions are usually inputs that impact the current behaviour of lower-level actuator control mechanisms, thereby impacting the estimated state of the world as training occurs.

However, other methods suggest addressing the reality gap with improved simulation software. For example, Ibarz *et al.* [Ibarz 21] reference techniques such as modelling or accurately replicating hardware-based parameters, including latency, dynamics and actuator models. Alternatively, domain randomisation has previously shown great success and remains a popular method. Domain randomisation alters dynamic parameters within the simulation to make a policy robust to such changes. If considering

visual aspects, one may alter texture and lighting features [Ibarz 21]. For example, a learning process will randomise friction, mass, and inertia parameters for actions interacting with the environment. An example of this approach is by Peng *et al.* [Peng 18], who also combine this approach with recurrent training methods to produce robust policies.

One also can observe other methods of simulation improvement. For example, Tan *et al.* [Tan 18] overcome the reality gap by modelling the actuators and latency responses of a Minitaur research robot. They also incorporated domain randomisation as part of their effort to overcome the reality gap. The authors note how domain randomisation, accurate actuator model and latency handling were essential in transferring locomotion skills directly to hardware. Alternatively, Collins *et al.* [Collins 20] evaluate different parameters that can assist in overcoming the reality gap. They detail how joint velocity and lateral friction heavily influence the reality gap and should not be excessively randomised. They also recommend running simulations at the default timestep set by the original developers, as a range of simulator parameters may be reliant on this optimised value.

2.3.4 Enhancing Learning

Improving Learning

While RL with SAC or TD3 can produce robust learned behaviour, there are augmentations to learning which can improve sample efficiency or stabilise training, such as batch normalisation, briefly introduced when discussing DDPG in Section 2.3.2. However, further examples are present in the literature. Demonstrating expert behaviour can improve sample efficiency as shown by Rajeswaran *et al.* [Rajeswaran 17]; their Demo-Augmented Policy Gradient (DAPG) algorithm improved sample efficiency and produced robust policies by incorporating human demonstrations into the training process. Zhu *et al.* [Zhu 19] used this method on manipulation tasks from the ROBEL benchmark, collecting kinematic motions before training with a natural policy gradient (NPG) approach. This method can assist in scenarios with sparse rewards. However, expert collection steps are required to apply such an approach.

Alternatively, one can observe steps to improve the sample efficiency by targeting transitions that provide the higher expected learning. For example, the prioritised experience replay (PER) approach [Schaul 15] assigns a weight to each transition in the replay buffer by estimating the TD error. Traditionally, the replay buffer applies to off-policy model-free approaches and consists of transitions with the tuple (s, a, s', r) . In the case of PER, this tuple takes an additional term resulting in $(s, a, s', r, |\delta|)$ with $|\delta|$ representing the magnitude of the TD error. PER calculates the TD error upon initially acquir-

ing a transition and recalculates $|\delta|$ every time the learning algorithm samples a transition. A sampling mechanism ensures that the transitions with a larger TD error magnitude are more likely to be sampled throughout training. Schaul *et al.* [Schaul 15] demonstrated a significant improvement in sample efficiency when using PER with DQN and double-DQN. However more hyperparameters require consideration when using PER.

Engstrom *et al.* [Engstrom 19] present a case study on two popular on-policy approaches of Trust Region Policy Optimisation (TRPO, [Schulman 15]) and PPO demonstrating that code-level-optimisations influence policy results more than the algorithm of choice. A similar study, including both on-policy and off-policy approaches [Henderson 18], assessed the impacts of hyperparameters, rewards, random seeding, evaluation methods, implementations, network architectures and environments. Findings by Henderson *et al.* [Henderson 18] include how hyperparameter selection and configuration are inconsistent across the literature, and how poor hyperparameter selection can be detrimental to training. Additionally, they discuss optimal activation functions for neural network approximators in RL. At a high level, they determined that ReLU (rectified linear unit) or Leaky ReLU usually performed the best. However, the research notes how such a phenomenon was inconsistent across algorithms and environments. Random seeding and trial numbers are other areas discussed, with an experiment [Henderson 18] demonstrating the wide variance different seeding can cause on an algorithm with static hyperparameters.

Henderson *et al.* [Henderson 18] also present an experiment demonstrating how significant reward scaling can impact training. However, they also show that layer normalisation can further impact any reward scaling. Such phenomenon, in part, can be attributed to how gradient-based methods react to large output scales, which can cause saturation and inefficiency in training procedures. In the original article on DDPG by [Lillicrap 15], batch normalisation (Ioffe and Szegedy [Ioffe 15]) was used to normalise inputs throughout training to generalise across task domains and rewards. This method allowed the authors to avoid manually scaling the features (including actions and rewards) for each task. The critical takeaway from Henderson *et al.* [Henderson 18] and Engstrom *et al.* [Engstrom 19] is that reward scaling, hyperparameters, normalisation techniques, network structure, activation operations, and code level improvements can play a significant role in algorithm success. Therefore, all integration details and learning configurations should be displayed when comparing approaches.

Reward Crafting

Reward creation is a crucial aspect of RL formulation. Ibarz *et al.* [Ibarz 21] discuss possible approaches to creating a reward function, including reward shaping, sparse rewards or learning the reward itself. Reward shaping is simply structuring a reward mechanism that guides a policy towards a goal. An example of such an approach is using task error as a negative reward. Ahn *et al.* [Ahn 20] use the error from the goal with augmentations for both a quadruped and manipulator robot to guide policies to minimal error conditions. Tasks involved include standing, orienting and walking for the quadruped, and manipulation tasks such as turning a valve for the manipulator. Using goal error is common in deep learning research tasks [Varin 19, Gu 17]. Such an approach, while effective, will not enable robots to generalise to other tasks and usually remains a limited or specific policy. However, it remains a viable method for learning certain skills on hardware platforms.

Alternatively, sparse rewards are another possible avenue where reward signals only occur on task completion. This scenario was investigated by Rajeswaran *et al.* [Rajeswaran 17] who presented DAPG to collect demonstrations before training the policies. This approach proved beneficial as the tasks addressed in this research included tool use, object relocation and environment manipulation with a 24 DoF robot hand. As rewards only occur sparsely, these tasks were complicated, and off-policy approaches such as DDPG failed to converge to a solution, whereas DAPG provided a dataset through which the agent could successfully learn behaviour. Akkaya *et al.* [Akkaya 19] combined sparse and shaped rewards in which a shadow hand performed dexterous manipulation and completed a Rubik's cube with a single hand. This approach used PPO with sparse rewards upon task completion or failure. For example, dropping the cube would penalise 20 points. Shaped rewards at each timestep reflected the error from the present state to the goal configuration.

Another example of sparse rewards [Kalashnikov 18] presented the QT-Opt framework, an RL framework targeting generalised grasping that uses a single reward to indicate success when grasping. The reward derives from an image subtraction method, by which the gripper drops a grasped object onto the workspace. If the gripper had been unsuccessful in the grasp attempt, then the workspace image would remain unchanged, and the image subtraction would reflect this unsuccessful attempt. Other approaches attempt to learn the reward function, also known as inverse learning. For example, Singh *et al.* [Singh 19] present a method providing a learning system a series of images representing successful completion. SAC then learns several tasks, including stacking books or draping a cloth over a box. Finally, the policy queries a human supervisor at a user-defined interval to determine if specific images

gathered throughout training represent a successful completion. The authors did not create a specific reward function; instead, the policy learned what states defined success from images while training.

Several components from the literature have altered traditional reward signals to be more adaptable to unknown dynamics and broader task scope. For example, Ma *et al.* [Ma 21] explored using space-time bounds in a reward function. The authors saw robust, trained policies on varying simulation tasks by constraining the acceptable error region and terminating the learning episode if boundary violations occurred. The authors also noted an improvement in sample efficiency. Furthermore, improvements occur when relying on space-time bounds for more complex tasks compared to traditional imitation reward schemes that do not necessarily always result in an optimal controller. One can also note how these authors approached feature extrapolation combined with space-time bounds, e.g. making a character perform a back-flip was achievable by dictating the position and orientation of the character's CoM.

Hindsight experience replay (HER) [Andrychowicz 17] presents a mechanism to learn from failure. For example, if an agent kicks a soccer ball and misses the goal by two meters under sparse reward conditions, traditional RL would not reward this behaviour or provide any indication that this sequence of events was partially correct. However, HER introduces methods that learn from failure and ask, what if the goal was two meters to the left? Would an improved reward be given under such a condition? The concept of HER is the integration of goals while training on a series of transitions, estimating rewards with the same trajectory but under multiple conditions. A set of goals, G , are defined such that the replay buffer takes an input of $(S_t, A_t, R_t, S_{t+1}, g)$ where $g \in G$. This modification diversifies the collected data to improve the training process. While it assists policies in finding sparse rewards, another application of HER is to provide some form of user-defined control over a policy. For example, the pushing task involves pushing a puck to the desired location on the table, which is an arbitrary location on the workspace [Andrychowicz 17]. One can also note that the goal parameter parses through the actor and critic networks.

Andrychowicz *et al.* [Andrychowicz 17] evaluate HER with a robot performing pushing, sliding or pick-and-place actions. They demonstrate how their method outperforms DDPG without HER as opposed to including HER. Interestingly, HER showed significant improvements in the original paper, where sparse rewards were present. However, in shaped reward problems, little to no improvement was observed. Two reasons, outlined by Andrychowicz *et al.* [Andrychowicz 17]. Firstly, the natural discrepancy between optimising shaped rewards and success conditions makes utilising such an approach questionable, and secondly, shaped rewards penalise inappropriate behaviour, hindering exploration. HER

can apply to both TD3 [Singla 20] and SAC [Lee 21].

Measuring Performance

The final aspect of RL to discuss is how agents measure success. How do researchers compare and validate that specific hyperparameters, reward formulations or algorithms result in the most robust agent or policy? At a high level, the answer is the policy that maximises the reward. For example, Meng *et al.* [Meng 21] evaluate their algorithms by performing the training process under four different seeds and then gathering the average of 10 evaluation procedures across a range of environments and state configurations. However, such an approach indicates how well an agent has learned to obtain a reward. As Henderson *et al.* [Henderson 18] and Engstrom *et al.* [Engstrom 19] discuss, this performance can be sensitive to hyperparameters, implemented code optimisations, random seeding or reward structures. If evaluating a new algorithm to existing SOA approaches, one should evaluate many environments and execute multiple training runs under different seeding as a minimum baseline approach. As code optimisations are often present in many different code bases, transparency regarding these features is essential when publishing results.

Henderson *et al.* [Henderson 18] specifically mention how these issues make simply detailing the maximum return, including averages across seeded trials, inadequate when comparing algorithms or discussing a training procedure. Instead, they recommend several approaches to validate a policy's performance, including bootstrapped confidence intervals. Such an approach gathers a wide range of evaluation episodes of a policy under random seeding and then sub-samples returns for several iterations to build confidence intervals of the policy's performance. Henderson *et al.* [Henderson 18] suggest such an approach as policy performance can vary extensively under random seeding. Additionally, they suggest a bootstrap power analysis to evaluate if a broader sampling process needs to occur.

Furthermore, Chan *et al.* [Chan 19] present various metrics for evaluating algorithm reliability. There are several elements to consider when measuring this aspect of an algorithm. For example, stable incremental learning is preferable during learning as opposed to noisy reactions to positive signals. Unstable learning can result in swings in performance and unpredictable deployment issues. Additionally, as previously noted, training across runs should remain stable. For example, training under various seedings, varying hyperparameters, and implementation details should still robustly learn policies. If they do not, then the algorithm will be unpredictable to various dynamics and may impact further applications and research. Finally, a learned policy after training should result in robust behaviour across multiple rollouts.

Chan *et al.* [Chan 19] detail statistical tests that enable robust evaluation of various RL algorithms against various environments. However, they also found that algorithms with the best median performance, usually TD3 or SAC, were not necessarily the most reliable given the metrics they developed. Such an observation remains in line with the previous statements of Ibarz *et al.* [Ibarz 21], who detail how these off-policy approaches can be unstable compared to their on-policy counterparts. Nevertheless, despite these possible intricacies, many robotic behaviours have successfully learned unique skills using SAC and TD3, even with simple evaluation procedures, as discussed in Section 2.3.3.

2.3.5 Conclusion

Reinforcement learning encapsulates the required technology for robots and advanced physical agents to learn policies that can execute specific tasks in simulation and hardware. At the same time, limitations remain, specifically in generalisation and adaptation to unforeseen circumstances. Nevertheless, robotic systems can learn behaviour that can overcome unknown dynamics and react appropriately to encountered difficulties. Furthermore, off-policy actor-critic algorithms have seen such outcomes achieved on low-cost robots. While the reality gap still presents significant challenges, domain randomisation and other optimisation techniques can enable learned policies to execute intelligent skills on hardware platforms.

2.4 Research Gaps

This literature review has surveyed topics surrounding robot fabric handling, including grasp characterisation, gripper design, and deep learning. In exploring these aspects, several research gaps have become apparent. Firstly, previous research has taken note of the advanced manipulation capabilities of humans when designing grippers or fabric manipulation strategies. As a result, some research projects designing manipulators replicate human morphological aspects or behaviours. However, these designs remain limited in application and capabilities, or contain complex features in order to provide generalised grasping capabilities. When exploring the literature, grippers that contained human-inspired elements did not comprehensively investigate human grasping of fabric from a hand-centric perspective. Such a study could lead to novel design inspirations and highlight previously ignored manipulation aspects. Therefore, the first identified research gap concerns a lack of complete understanding regarding anthropomorphic hand-centric fabric manipulation. Chapter 3 addresses this limitation by surveying previous

grippers with anthropomorphic hand-centric taxonomies to discover novel unique aspects in design and behaviour, these observations are then validated with a user study which observes humans performing a range of fabric handling tasks. This results of this first investigation inspire the conceptual gripper design, providing the groundwork for Chapter 4.

A simple yet generalised manipulator for textile waste sorting does not exist. This observation is the second gap identified from the literature review. This thesis takes the conceptual gripper formulated from the anthropomorphic study of Chapter 3 and creates a prototype using established modelling techniques and a fabrication process. The fabricated device improves upon the SOA by remaining a serial-link manipulator with four actuators and no complex components, while offering a combination of grasping skills previous devices do not. Finally, the remaining research gap concerns the act of environmentally constrained grasping. Grasping while exploiting environmental constraints is an essential skill that humans use as part of their diverse manipulation capabilities, and some grippers from the literature target such a skill. However, present manipulators that target this capability can only perform such an act from specific positions, and usually with pre-programmed trajectories. The anthropomorphic survey of Chapter 3 describes specific limitations in environmentally constrained grasping capabilities of previous devices. Grasping in this manner is a complex, contact-rich skill that can become more complicated depending on the nature of the interaction and the environment dynamics. An unexplored avenue to overcoming these complexities is the use of reinforcement learning. The exploration of Chapter 5 addresses the translation of observed human behaviour into a reward function.

Chapter 3

A Human-Inspired Investigation of Dexterous Fabric Manipulation

3.1 Introduction

Designing end-effectors for fabric manipulation remains a complex challenge involving parameters such as the desired application, intended capabilities and technical feasibility. As outlined in Section 2.2.2, previously developed devices remain limited in functionality, and do not generalise to broader applications. In addition to limited applicability, previous grippers implement technical augmentations to assist grasping behaviour. For example, the Mag-Gripper [Marullo 20] embedded an electromagnet in the fingertips to attract metal components within the fabric while grasping, and the roller-fingertip device [Abe 20] used actuated rollers in contact with fabric to pull material into a grasp. Alternatively, a lamprey-inspired manipulator [Ku 20] used a combination of microneedles and a vacuum mechanism to assist with delicate fabric-handling tasks, while other devices used tendon-driven components [Koustoumpardis 14, Von Drigalski 17b]. Mechanically compliant joints are another observed complex design feature present in the grippers of Le *et al.* [Le 13] and Donaire *et al.* [Donaire 20].

As existing devices remain limited in application, or embed complex components into their design, recent research has attempted to qualify manipulation primitives within robotic fabric manipulation to inform gripper design scope and outline desired manipulation skills. Borràs *et al.* [Borràs 20] present a framework for defining grasps of fabric from an object-centric perspective by studying previous devices. Such a framework enables researchers to outline desired grasp contacts and inform the required geometries of novel grippers. This framework was instrumental in developing the gripper presented by Donaire

et al. [Donaire 20]. Borràs *et al.* [Borràs 20] specifically refer to their framework as necessary, given that existing grasp taxonomies are insufficient for characterising textile manipulation. Such an assessment particularly applies to object-centric definitions, as fabric displays significantly different physical characteristics when compared to static objects. However, while the object-centric perspective can assist in informing manipulator development, it does not directly correlate to a gripper's behaviour and ability, as Bullock *et al.* [Bullock 12] mention. Bullock *et al.* [Bullock 12] further note that hand-centric studies can directly compare the capabilities of different grippers and formalise manipulation primitives. Viewing grippers under an anthropomorphic hand-centric lens can provide a viewpoint that inspires gripper design through formalised manipulation definitions and human morphological features. A common trend in the literature is that specially designed grippers take anthropomorphic inspiration from human morphology or behaviours. However, these projects usually took heuristic human observations to justify a design approach [Koustoumpardis 14, Von Drigalski 17b, Shibata 09].

Considering the human-inspired designs from previous research, and the lack of hand-centric gripper studies within fabric manipulation, this chapter poses the question; '*Can discussing previous grippers from a hand-centric, anthropomorphic viewpoint reveal unique limitations and highlight novel design inspirations?*' This chapter refers back to the *Double Diamond Design Thinking* framework from Section 1.4 [Herath 22] to address this query. This framework provides a methodology to derive novel and optimal solutions for indeterminate problems or gaps using the steps of *discover*, *define develop*, and *deliver*. A discovery process has already occurred in Chapter 2, identifying the need for a generalised robot manipulator solution that can apply to pick and place sorting of diverse textile waste streams. In addition, the *discover* phase identified a research gap where no previous grippers had thoroughly investigated robotic fabric manipulation using hand-centric classification schemes.

Existing anthropomorphic hand-centric grasp and manipulation taxonomies use neuroscientific, biomechanical, and robotic definitions to formally describe hand-centric behaviour. Early definitions for broad grasp characterisations came from the medical publication of Napier [Napier 56], who initially presented the power or precision classification. Furthermore, the neuroscientific publications of Iberall *et al.* [Iberall 86, Arbib 85, Iberall 97] investigated human grasping to further develop concepts that could refine grasp classification. Cutkosky [Cutkosky 89] used these concepts to develop a classification scheme for human manipulation behaviour that could translate to robotic agents. More recent publications [Feix 15, Bullock 12] use these concepts from varying fields to define anthropomorphic manipulation configurations and grasping motions while focusing on robotic applications.

The core activity of this chapter uses two state-of-the-art taxonomies to study existing grippers and construct a comprehensive understanding of dexterous manipulator skills required within fabric manipulation. Such an understanding will lead to a generalised manipulator suitable for textile waste sorting. The neuroscientific grasping definitions come from the GRASP taxonomy of Feix *et al.* [Feix 15], while the IHDM taxonomy of Bullock *et al.* [Bullock 12] defines anthropomorphic manipulation primitives. Under the Double Diamond development scheme, this investigation falls under the *define* step, which collects data to inform a novel solution. By observing a subset of described grippers, this chapter builds fabric-specific versions of the GRASP and IHDM taxonomies to describe the grasp poses and dexterous actions of previous robotic grippers manipulating textiles. In addition, these details are mapped to the surveyed devices while discussing further information, including applied grasp forces and technical details. The *define* step concludes by discussing novel insights derived from this anthropomorphic hand-centric survey.

The *develop* step of the double diamond approach splits into the substeps of *ideate* and *prototype*. The ideation process uses observations from the anthropomorphic, hand-centric discussion to identify present limitations and previously unexplored grasping solutions. Given the identified gaps, this chapter formulates a novel anthropomorphic design that addresses previous limitations. Unlike previously described devices, the proposed gripper derives from a single anthropomorphic configuration within the GRASP taxonomy, the *lateral* grasp. This grasp is a simple configuration that only uses the thumb and side of the index finger to formulate a grasp. A single serial-chain robot structure can replicate this configuration as Chapter 4 demonstrates. The hypothesis in formulating such a device is that a human can perform all the necessary dexterous skills to manipulate fabric while constrained to this grasping configuration.

A user study is presented that asks human participants to manipulate garments under various kinematic constraints to validate such a hypothesis. As the study continues, instructions to participants limit aspects of dexterous manipulation allowed while handling the fabric. The results validate that the lateral grasp is sufficient for the desired waste sorting applications. A discussion of further observations from the neuroscientific and psychological studies is also present before outlining the final design to model and fabricate. This study and discussion conclude the *ideate* phase of the design thinking process. Chapter 4 then addresses the *prototype* sub-step, which develops iterations of the novel gripper while refining the design. The prototyping process also leads into the *deliver* step, which deploys the manipulator onto a robotic arm within the HCT robotics lab at the University of Canberra.

3.2 Manipulator Selection

The grippers surveyed throughout this chapter are from an identified subset of the literature that fulfils a specific criterion. Selected devices come from academic research projects that built grippers explicitly designed for fabric manipulation, and they all have in-hand dexterous capabilities beyond simplistic grasping, including haptic exploration, environmentally constrained (EC) grasping, or grasp gaiting. These constraints result in the listed manipulators of this section excluding commercial manipulators. As Borràs *et al.* [Borràs 20] and the literature discussed in Section 2.2.2 highlight, commercial or general grasping devices applied to intelligent robot garment handling were limited to only using point-to-point grasps and lacked the features to execute more refined in-hand dexterous behaviours. Section 2.2.2 elaborates on how dexterous intrinsic and extrinsic skills are essential in the complex challenge of generalised fabric manipulation, further cementing the logic for omitting commercial devices limited to basic grasping.

In addition to these conditions, grippers with technical augmentations to assist with grasping behaviour are also omitted from the survey. This requirement prevents gripper designs that use non-anthropomorphic grasping mechanisms from being included in this discussion. Examples include devices with electromagnets to attract metal components [Marullo 20, Dragusanu 22], microneedles with a suction mechanism inspired by the lamprey fish [Ku 20], or roller fingertips with a brush-like coating to assist with the grasping of flattened materials [Abe 20, Yamazaki 21]. As the purpose of this survey is to qualify previous grippers designed for fabric manipulation under human-centric definitions, the inclusion of devices with these complex features would be problematic to describe with the taxonomies of Bullock *et al.* [Bullock 12] and Feix *et al.* [Feix 15]. Additionally, technical augmentations, including the examples given above, can increase the design complexity and following the discussion of Section 2.1, the research of this chapter assumes that human-inspired designs are best suited to discovering a novel generalised grasping solution.

Considering the given constraints, a total of eight research projects aligned with the desired scope of discussion were identified. While the scope of these eight projects surveyed can initially seem limited, this survey aims to establish the requirements of a generalised human-inspired gripper solution. Focusing on the directly relevant grasping skills and human-comparable grippers can establish unique gaps and design scope. The following subsections present a description and figure of each device. However, some authors do not provide comprehensive information, including applicable grasp forces and limited modelling information, as several projects target different applications and do not consider the full scope

of fabric manipulation. This section makes an effort to include all relevant information about each device. The list below assigns an integer to each manipulator as a reference throughout the chapter.

1. CLOTHILDE Project Gripper [Donaire 20]
2. CloPeMa Gripper [Le 13, Le 15b]
3. 3 Fingered Under-Actuated Gripper [Koustoumpardis 14]
4. NAIST M2S Openhand Gripper [Von Drigalski 17a, Von Drigalski 17b]
5. Cloth Gripper Based On CAM Mechanism [Koustoumpardis 17]
6. 3 Fingered Anthropomorphic Gripper [Ono 01]
7. Dual Pincers Prismatic Gripper [Shibata 12]
8. Prismatic Edge Tracing Based Grippers [Sahari 10]
 - 8.1 Basic Gripper
 - 8.2 Roller Gripper
 - 8.3 Inchworm Gripper

3.2.1 Gripper 1 - The CLOTHILDE Gripper

The most recent device published, Gripper 1, was presented by Donaire *et al.* [Donaire 20] as part of the European project CLOTHILDE. The author's inspiration for this device came from the limited generalisation of previous manipulators developed for fabric manipulation. In particular, the authors discuss how many previous devices can only perform pinch grasps mimicking a finger(s) and thumb structure. They acknowledge that two-fingered pinch grasps can perform various tasks, including grasping crumpled clothes or manipulating garments bi-manually to lay out and fold materials. However, Donaire and their colleagues argue that more refined manipulation actions require a broader range of dexterous skills. Thus, they targeted moving folded clothes, folding a T-shirt, tracing an edge and more generalised manipulation. In order to construct this manipulator, they created an initial prototype controlled by a human user with rubber bands for actuation to evaluate a series of desired grasp configurations and geometric properties.

The resulting device was a two-fingered manipulator that could split the lower finger into two sub-fingers (an abducted state), creating a plane-like structure within held garments (see the lower image in Figure 3.1). The lower finger also had compliant joints to safely collide with the environment with a thin structure to slide underneath clothing on a surface. Two Dynamixel XM430-W210 servo-motors actuated motions within the fingers. Finally, the upper finger contained a contact friction alteration mechanism based on the device described by Spiers *et al.* [Spiers 18]. A silicon pad could retract and extrude from the fingertip, increasing friction between the fabric and the manipulator. Such a feature increases grip strength via friction when extruded. If retracted, a reduction in friction enables the device to slide across a fabric's body in a grasp gaiting act. Donaire *et al.* [Donaire 20] also discuss how the manipulator is ill-suited to rigid objects. Thus, they mention further plans to add additional movement to the thumb appendage (f_1 in Figure 3.1). Device evaluation occurred with human operators performing arm motions while the gripper executed dexterous acts. A grasp strength evaluation also took place by holding fabrics with various additional weights.

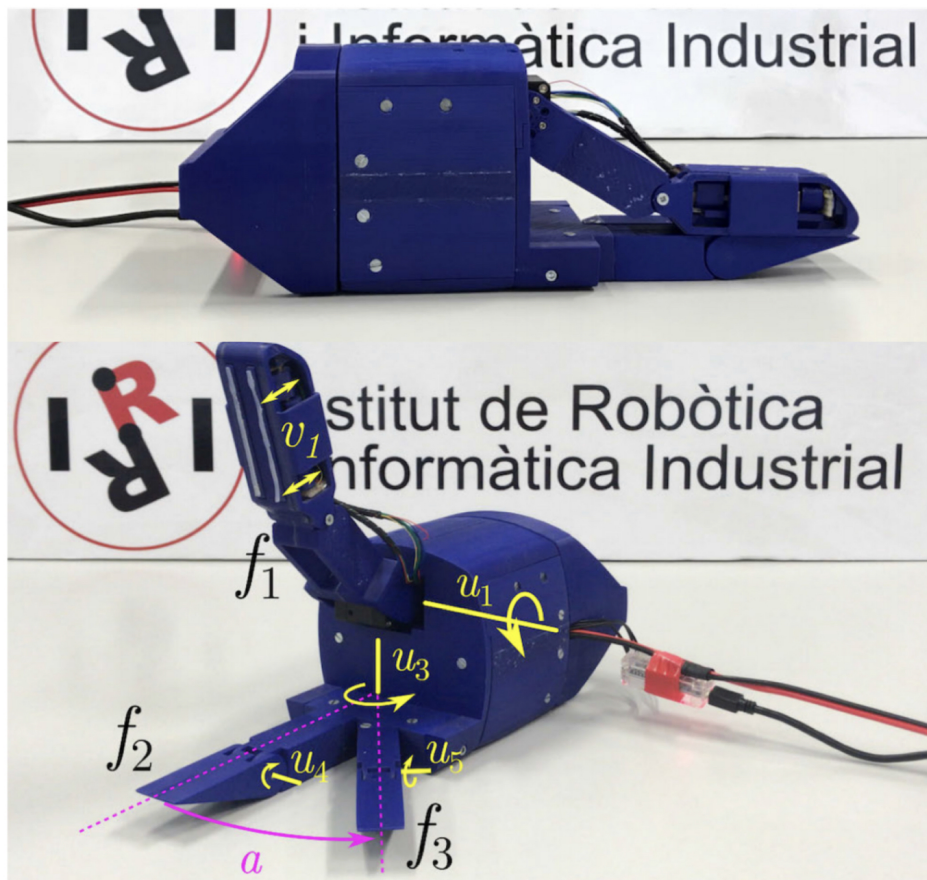


Figure 3.1: Gripper 1 [Donaire 20]

3.2.2 Gripper 2 - The CloPeMa Gripper

Another gripper from an earlier European research project, Clothing Perception and Manipulation (CloPeMa), was presented in a range of articles by Le *et al.* [Le 13, Thuy-Hong-Loan Le 13, Le 15b]. The CloPeMa project focused on manipulating garments, emphasising recognition, unstructured picking/extraction of garments, and garment folding with applications in a domestic setting. Therefore, one area of investigation within the CloPeMa project was developing a new robot gripper design that could fulfil these requirements, henceforth referred to as Gripper 2. Regarding considerations within the design, the authors outline navigation of environmental constraints, geometric considerations, and target items for grasping and grip strength. In order to address environmental constraints, a variable impedance actuator controls the stiffness of compliance in the fingers during operation. They also mention how simplicity of the manipulator was a primary objective. Thus the overall structure mimics a precision grasp inspired by an index thumb grasp pose. The proposed mechanism and modelling process for the variable compliance of the fingers is presented by Le *et al.* [Le 15b]. From the outset, the authors also required Gripper 2 to perform haptic exploration with a rubbing motion. Other considerations in the design include the geometric proportions of the device, ensuring that the manipulator could, for example, hold materials of varying thicknesses with appropriate spacing configurations.

Additionally, the fingers needed to be thin enough to slide under held materials while performing EC grasping. While EC grasping, the compliance in the fingers would increase to navigate collisions with the surface while the thin finger appendage slides under an edge of the fabric. In order to gather information about the targeted garments for manipulation, the creators of Gripper 2 develop a ‘reference laundry

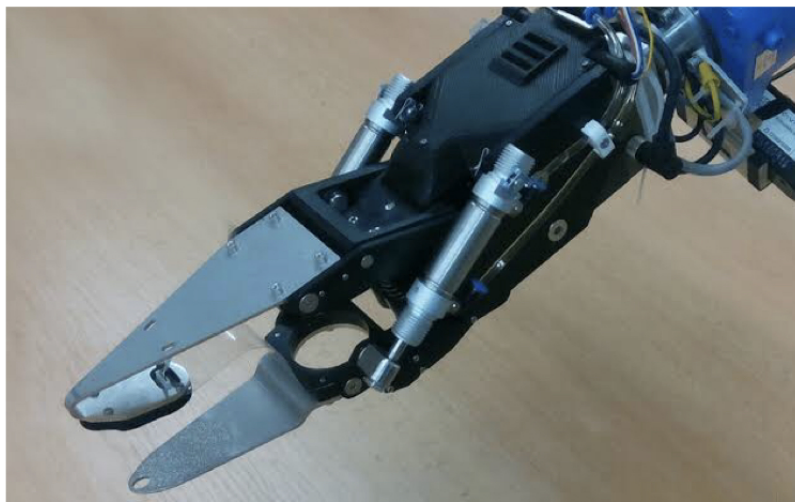


Figure 3.2: Gripper 2 [Petrík 15]

heap' based on sales data to benchmark target manipulation objects. Additionally, Le *et al.* [Le 13] scope extraction and grasping forces to ensure the device has an adequate grip strength for generalised cloth manipulation. To scope these requirements, Le and colleagues conducted a human-observation study where participants used a glove with force sensors and held garments to inform the strength requirements for grasping clothing generally and in entangled conditions. The key takeaways from such a study found that for an object weighing $5N$, one requires a maximum extraction force of $35N$ alongside a maximum grasp force of $30N$. The resulting manipulator derived from these requirements was an industrial device visualised in Figure 3.2. Finally, a sensor at the fingertip consisted of a capacitive pressure sensor array, microphone and ambient light sensor. The microphone and pressure array interpret haptic exploratory actions, and the ambient light sensor validates grasping success [Denei 17]. Publications by Le *et al.* [Le 13, Thuy-Hong-Loan Le 13, Le 15b] elaborate on the design, modelling and evaluation procedures for further information.

3.2.3 Gripper 3 - Three Fingered Underactuated Manipulator

Gripper 3 was introduced by Koustoumpardis *et al.* [Koustoumpardis 14]. The device broadly mimics the anthropomorphic structure of the human hand with a palm, thumb, index finger and middle finger. The authors sought to replicate human-like manipulation primitives and created a manipulator with several anthropomorphic features. The targeted application of this manipulator was to perform EC grasping. Koustoumpardis *et al.* [Koustoumpardis 14] reference behaviour where the thumb pins fabric to a surface before using the index and middle fingers to drag along the environment surface, deforming fabric into

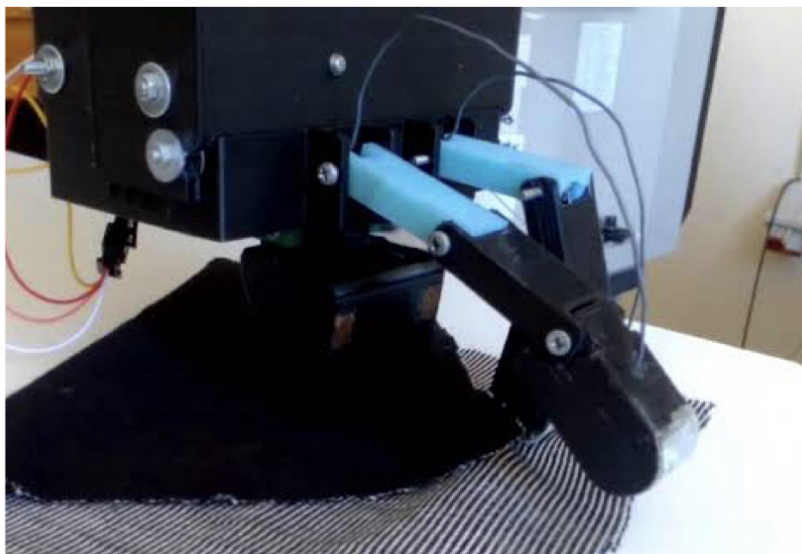


Figure 3.3: Gripper 3 [Koustoumpardis 14]¹

a grasp. The developed prototype was a three-fingered manipulator that used the non-actuated thumb to detect collisions with the environment surface, alongside two fingers actuated via tendons to HS-85MG servo-motors hosted within the main chassis. The authors present kinematic and kinostatic analyses alongside fabrication details in their article.

The thumb component only contained a single compliant joint, using a torsional spring, to safely collide with the environment surface before EC grasping took place. Torsional springs were also present in the tendon-actuated components to define a default position. In order to detect successful grasps, metallic laminates acting as insulators were present in the thumb and finger components as part of an electric circuit. Such an approach was practical for their grasping experiments, which involved grasping flattened fabric in five scenarios. While performing this experiment, the authors saw that the motions fail when grasping a material with low friction. Such phenomena could occur for several reasons, including pre-programmed trajectories that exert an inadequate normal force on the table or a low friction factor between the manipulator and the target fabric. Koustoumpardis *et al.* [Koustoumpardis 14] suggest placing silicon overlays on the fingertips of the gripper to increase the friction between the hand and garments.

3.2.4 Gripper 4 - The NAIST M2S Openhand

The NAIST M2S OpenHand, Gripper 4 [Von Drigalski 17b], was a precision type gripper with two fingers. The design extended the model M2 design [Ma 16] at the Yale GrabLab. A unique aspect of this device was the innate ability to switch between modes of under-actuation [Ma 16]. Gripper 4 represented a precision grasp structure with a static thumb and a two-link finger actuated by two tendon-driven servo-motors. The creators of this gripper outline several goal capabilities of tucking fabrics into small spaces (e.g., as required in bed-making), active perception to determine grasp success, and dexterous grasping alongside in-hand manipulation. In addition, the device could sustain substantial axial loads on the static thumb appendage for tucking fabrics. Other capabilities included tactile feedback, relative sensor motions and flat fingers [Von Drigalski 17b].

In order to determine grasp success, a triaxial force sensor was present on both fingertips of the manipulator. Thus, a rubbing motion inferred friction information from the sensors, determining grasp success. In addition, Gripper 4 performs three basic manipulation primitives of grasping, sliding (grasp gaiting) and tucking. The authors also mention increasing the friction factor on the gripper surface if

¹Figure 3.3 was sourced from the associated media of the article by Koustoumpardis *et al.* [Koustoumpardis 14].



Figure 3.4: Gripper 4 [Von Drigalski 17b]²

heavier objects require manipulation. Finally, the authors evaluate the device by observing the grasp forces at various actuator positions and tracking the feedback of the force sensors while performing haptic exploration. Some limitations of this article are outlined by the authors, including errors in sensor positions while tracking grasp strength, and the inability to perform EC grasping. Future research avenues suggested by the authors included modifications to enable EC grasping and automation of the tucking task.

3.2.5 Gripper 5 - CAM-Follower Mechanism

Based on the CAM-follower mechanism, Gripper 5 was a second gripper produced by Koustoumpardis *et al.* [Koustoumpardis 17]. Like Gripper 3, this manipulator is a three-fingered device designed to grasp fabric in a flattened state using the environment. The authors outline several required features of this manipulator for grasping flattened clothing in a range of states, i.e. grasping from corners, edges, and the centre of the fabric's body, or grasping in 'folded-over' configurations. Additionally, they require this effector to be small enough to go on a mobile robotic system and to be actuated by a single servo motor. From an anthropomorphic viewpoint, the creators describe this manipulator as a thumb and middle finger pair that move vertically to pin the fabric, while the index finger in the centre rotates to grasp the material. As this system only uses a single rotational actuator, all fingers are attached to a single shaft with CAM mechanisms that enable movement of the fingers. Koustoumpardis *et al.* [Koustoumpardis 17]

²Figure 3.4 was sourced from the associated GitHub repository, <https://github.com/naist-robotics/naist-openhand-m2s>, of the article by Von Drigalski *et al.* [Von Drigalski 17b].

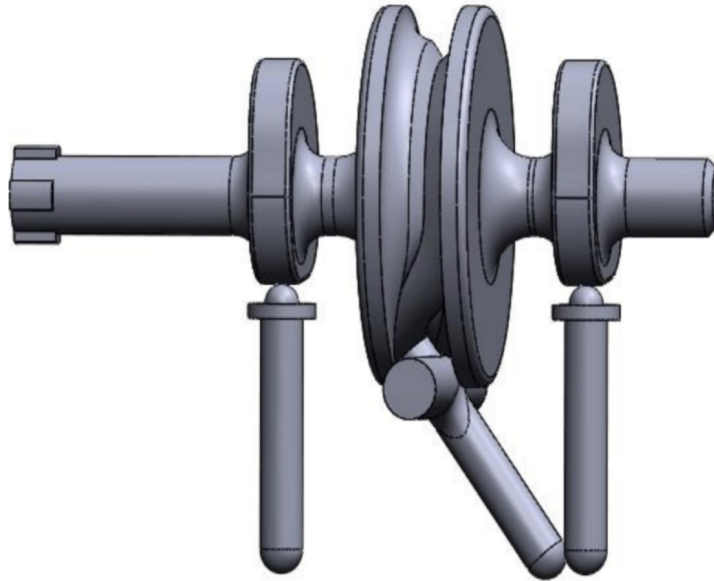


Figure 3.5: Gripper 5 [Koustoumpardis 17]

outline some preliminary kinematics of the manipulator alongside an evaluation procedure. The final manipulator was a 3D printed device that used springs to make the pinning fingers compliant alongside undefined overlays to the fingertips. A preliminary evaluation occurred where the device attempted to grasp various materials while performing acts including EC grasping and unfolding.

3.2.6 Gripper 6 - Three Fingered Anthropomorphic Manipulator

Gripper 6 was an anthropomorphic device [Ono 01, Ono 05] also utilised in a study by Ono and Takase [Ono 07]. This gripper consisted of a middle finger, index finger, palm and thumb, making it the most complex device in terms of actuators and degrees of freedom. Per images presented in the original description [Ono 01], the size of this device is comparable to a human adult's hand, albeit with some proportional differences. The authors model this device as several serial-link structures where the distal interphalangeal joints (the 'final' joints) in the non-thumb serial links do not independently actuate but passively link with the previous joints for actuation. In total, this gives the system nine degrees of freedom (DoF) with seven actuators.

Due to Gripper 6 drawing significantly from anthropomorphic inspirations and the numerous DoF, this device remains capable of a diverse range of anthropomorphic poses, thus allowing the manipulator to grasp fabric with various motion strategies. In the description of Ono *et al.* [Ono 01], the authors target the extraction of fabrics laid on top of each other. They note the complexity of this task as involving contact friction, regular forces between the fabric layers, and the manipulator's interaction with the top

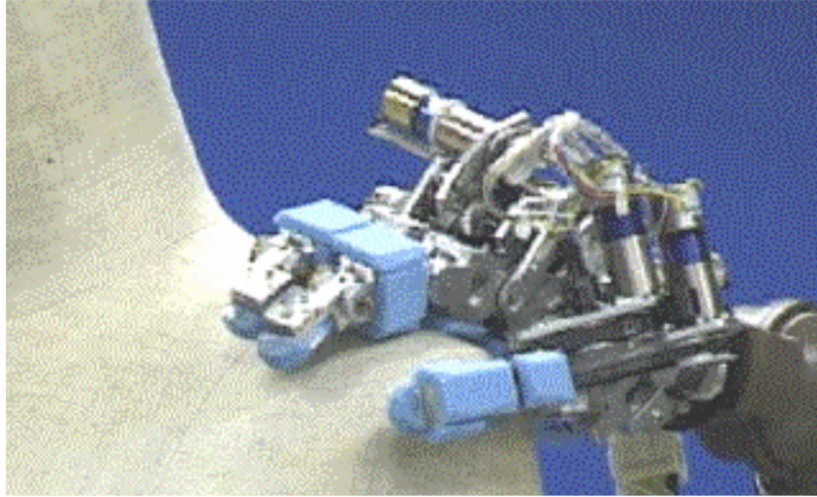


Figure 3.6: Gripper 6 [Ono 07]

layer. They also highlight how the resulting fabric configuration after open-loop robotic manipulation can result in substantially altered configurations. Finally, they evaluate two motions, in which the index finger drags inwards into the palm for a grasp, and another drags the thumb into the side of the index finger. Ono and Takase [Ono 07] expand on this research by further studying ideal motions and points on the fabric to grasp.

3.2.7 Gripper 7 - Prismatic Sliding Manipulator

A series of articles by Shibata *et al.* [Shibata 08, Shibata 09, Shibata 12, Shibata 16] introduce and use Gripper 7, which consisted of two sub-grippers, or clamps, on a long prismatic actuator. The target application of this device was to perform EC grasping, unfolding, and a specific spreading-out motion for placing material on a surface. One can also note that bimanual manipulation skills inspired these target applications for this single gripper. This actuator had 4 degrees of freedom, with the two clamp grippers moving vertically for grasping and releasing fabric between the fingertips; each clamp gripper could also move horizontally along the rail. To execute EC grasping, the manipulator would move the clamps to the outer extremes of the prismatic rail before using pre-programmed trajectories to place the clamps against the flattened fabric on the table. Then, the clamps would slide inwards while in contact, creating a protrusion with the fabric. The authors also embed rubber pads on the lower surface of the clamp devices to increase friction between the fabric and the manipulator.

The authors also establish pre-programmed traversing length parameters that allow the manipulator to perform EC grasping motions that result in a stable and predictable protrusion. However, these values

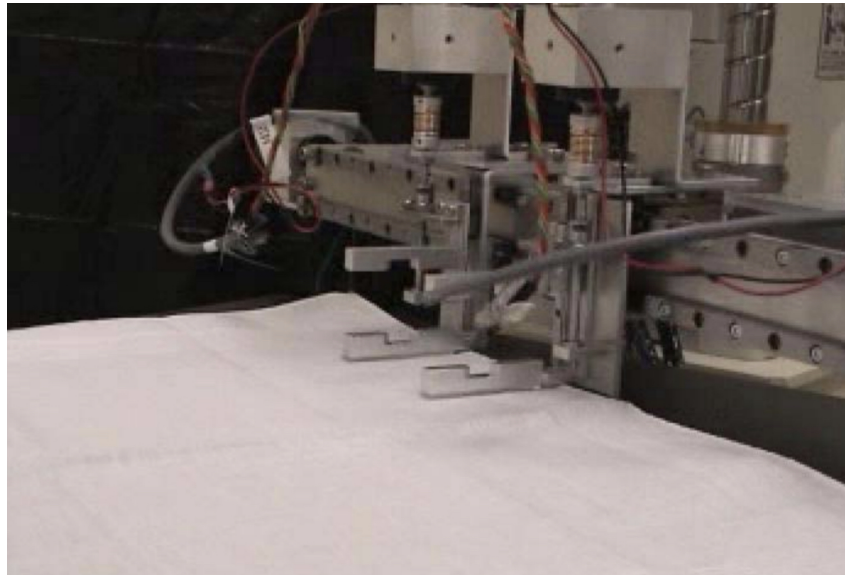


Figure 3.7: Gripper 7 [Shibata 12]

are derived manually for individual fabrics. Such an approach will not be generalisable to a broader context when using materials with diverse mechanical properties. When unfolding, Gripper 7 grasps the protrusion with both clamps and then performs a grasp gaiting action by moving outwards along the rail, letting the fabric slide between the clamped grasp. It remains unclear what grasp force was applied to enable grasp gaiting. Finally, an exploration where this device applied a ‘placing’ motion was assessed [Shibata 12]. However, this task did not influence the overall design. Gripper 7 is also physically the most prominent device of the described subset of grippers, with a width of approximately $742.2mm$. Thus, the strategy for grasping and unfolding could almost be considered a bi-manual manipulation with multiple grasp points and two sub-grippers within the system.

3.2.8 Gripper Set 8 - Garment Traversing Manipulators

The series of grippers, **8.1**, **8.2** and **8.3**, were developed by Sahari *et al.* [Sahari 10] who focus on the task of edge tracing or grasp gaiting. Each device was modified to perform grasp gaiting in a slightly different manner. Gripper **8.1** is the ‘basic gripper’, which remained a simple open-close device controlled by a servo-motor. This device was capable of a grasping force of approximately $5.472N$ and contained strain gauges at the fingertips to measure grip strength. Infrared sensors were also present to validate if a fabric was within the fingertips. Secondly, Gripper **8.2** is the ‘roller gripper’, being a device that expands on Gripper **8.1** by integrating roller fingertips and what the authors describe as ‘inner fingers’. The roller fingertips are spur gears freely rotating on a roller bearing to assist grasp gaiting, while the inner fingers are present to hold clothing securely. Both strain gauges and infrared sensors are present. This variant

has a reduced grasping strength of approximately $0.93N$. Finally, Gripper **8.3** is the ‘inchworm gripper’. This device is significantly different to grippers **8.1** and **8.2**, as it contains two sub-grippers or clamps, one of which is static in position while the second can move along a prismatic rail (Figure 3.8). Additionally, this device has three actuators: the first moves one of the clamps along the rail, and the others operate the opening and closing motion of both the sub-clamps. The two clamp devices retained strain gauges and infrared sensors, and this manipulator could grasp with a force of approximately $3.33N$. The edge tracing task targeted in this research involves a second robot holding a point known as the ‘first corner’, and these manipulators must perform grasp gaiting to move to the ‘second corner’.

Gripper **8.1** performs this task by applying a user-defined force on clothing as it slides the fabric between the fingertips to reach the desired configuration, with feedback from a vision system and the sensors in the fingertips. However, the authors identify an issue where Gripper **8.1** cannot backtrack along a grasp gaiting trajectory due to complications related to the fabric’s configuration or gravitational forces. Gripper **8.2** performs a similar action, however the roller fingertip reduces the friction between the fingers and the fabric, enabling this manipulator to backtrack along the trajectory if necessary. While Gripper **8.2** improves the manipulation task over Gripper **8.1**, the clothing can still fall away from the grasp.

The design of Gripper **8.3** combats this issue by allowing the manipulator to grasp clothing firmly while performing grasp gaiting. Such action occurred in an ‘inchworm’ fashion by holding the clothing with one clamp, opening the second, sliding along the clothing before grasping, moving the clamps together and then repeating this ‘inchworm-like’ action. The authors note how this technique eliminates the need for force control. In order to evaluate the devices, the robots manipulate a single square piece

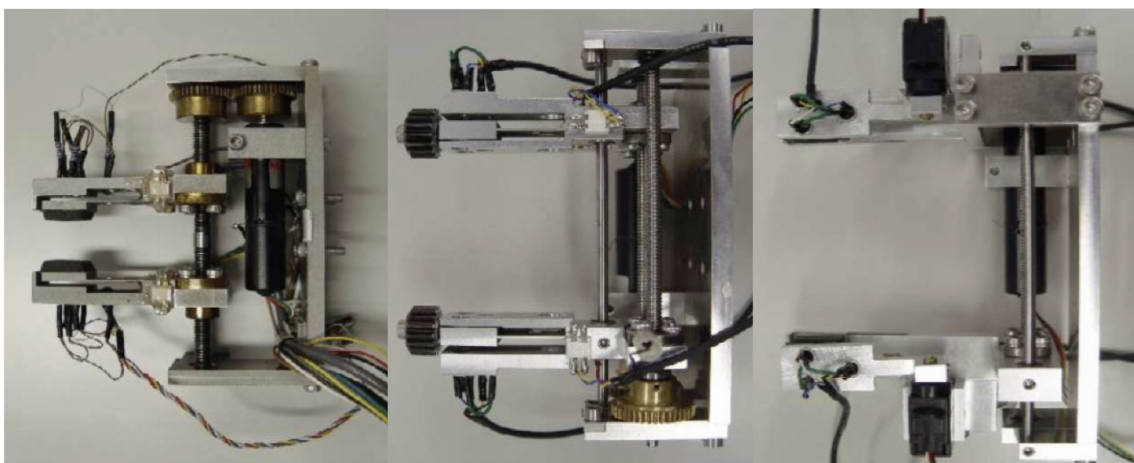


Figure 3.8: Grippers **8.1**, **8.2** and **8.3** [Sahari 10]

of fabric with all three grippers performing the grasp gaiting motion 30 times. The authors saw that Gripper **8.3** returned the highest success rate. However, each device used a slightly modified algorithm to perform the manipulation, possibly influencing the results of the manipulation comparisons.

3.2.9 Final Remarks

The manipulators described here present a range of capabilities beyond the simple grasping of fabric. For example, some target in-hand dexterous skills, while others target grasping motions to exploit the environment. Further than these discussed skills, one can observe that various grasping forces and mechanical elements are present. An overarching topic of interest in these grippers is the diverse approaches to EC grasping. For example, some use compliant mechanical setups to collide with the table safely, while others use pre-programmed trajectories with various methods of modifying the friction between the fabric and the manipulator. Another point of interest was how these grippers highlight the role of friction within in-hand dexterous manipulation and EC grasping of fabric.

3.3 Taxonomies for Anthropomorphic Manipulation

In order to establish a consistent description of the grippers' capabilities, anthropomorphic hand-centric taxonomies are used to classify behaviours and grasp configurations. These anthropomorphic frameworks derive from the GRASP taxonomy [Feix 15] for grasp pose comparisons and the IHDM taxonomy [Bullock 12] for dexterous manipulation actions or sequences. Whereas Chapter 2 gave a brief overview of these taxonomies, the following content presents an in-depth description and outlines their applicability to formalising dexterous manipulation primitives that qualify both the grasp pose and dexterous actions of manipulators. However, the original taxonomies contain elements that do not apply to fabric manipulation. Therefore, this chapter creates smaller, fabric-specific versions of the GRASP and IHDM taxonomies which utilise only a required subset of grasp poses and manipulation capabilities.

3.3.1 Addressing Grasp Poses

Parameters for Grasp Definition

Defining the parameters by which one can make anthropomorphic comparisons to robotic manipulators remains complex, with the mechanical designs of the evaluated grippers differing significantly from the human hand both functionally and physically. However, abstract anthropomorphic hand-centric defi-

nitions allow one to make comparisons to grasp poses regardless of the physical characteristics of the robotic manipulator. At a high level, one can classify any grasp as a precision or power grasp. For power grasps, an increase in surface contact and a static pose are vital characteristics. Alternatively, precision grasps are refined and capable of sensitive in-hand dexterous manipulation. Examples of these grasp types include holding a cricket bat at the handle for a power grasp, and writing with a pen for a precision grasp. Finally, some grasps contain elements of both power and precision, also known as an intermediate grasp. These three options of power, intermediate and precision are used in the GRASP taxonomy as the first of four elements to classify anthropomorphic grasp poses.

The second concept is Virtual Fingers (*VF*), being abstract representations of hand surfaces providing an opposition force per a definition provided by Iberall [Iberall 97]. Depending on the grasp configuration and direction of forces applied within the grasp, defining *VFs* can occur at the palm, a group of fingers or single digits. The third element relates to the direction of force applied within a grasp, called the opposition type. When one applies a prehensile grasp to any object, there are usually at least two *VFs* involved in the interaction exerting forces against each other. Opposition types refer to a feature that can describe the direction of force a grasp holds relative to the palm (usually represented with a coordinate frame) and three possible labels, as listed below.

- Pad opposition - Referring to a grasp between the thumb and fingers where contact occurs parallel to the palm, see Figure 3.9A.

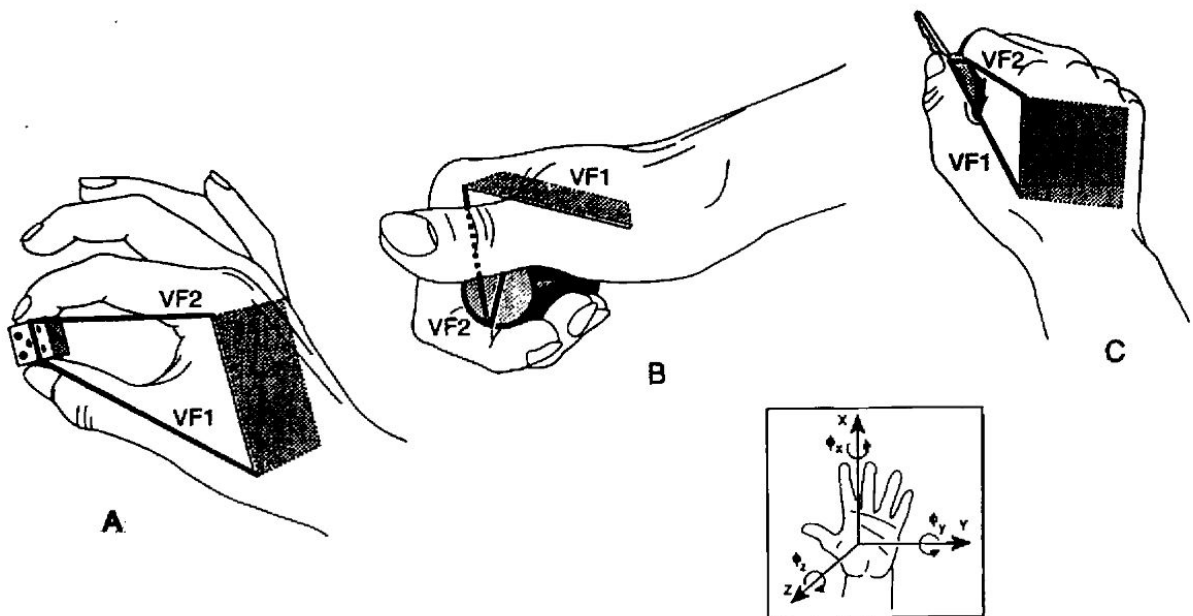


Figure 3.9: A visualisation of VF and Opposition Types, Iberall [Iberall 97].

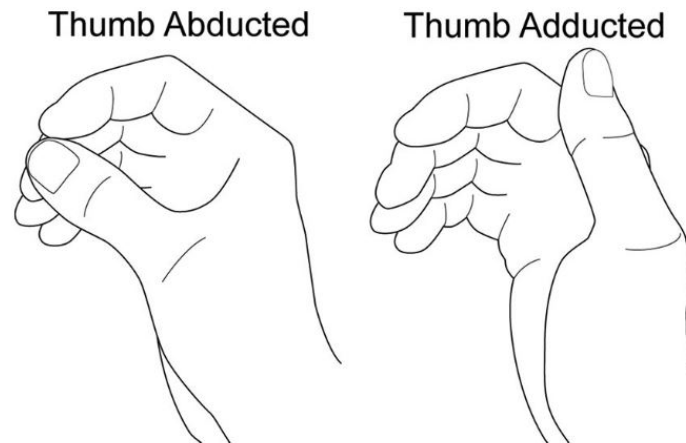


Figure 3.10: The human thumb in an abducted (left) or adducted (right) position [Feix 15].

- Palm opposition - Referring to a grasp between the fingers and palm. Where contact is perpendicular to the palm, see Figure 3.9B.
- Side opposition - Defining a grasp where contact points on the hand surface are in directions transverse to the palm, see Figure 3.9C.

These principles and grasp types were published in various descriptions throughout the 1980s by Iberall and colleagues. Iberall *et al.* [Iberall 97] provides a succinct summary of these concepts and their applications, and Figure 3.9 displays a visualisation of both *VFs* and opposition types. The fourth element used by the GRASP taxonomy is the feature of thumb adduction/abduction. The rotation of the carpometacarpal (CMC) joint in the thumb appendage can be classified as either adducted or abducted; Figure 3.10 provides a visualisation of these positions. Feix *et al.* [Feix 15] provide the first known example of using the thumb rotation as a classifying feature and, along with the previously described features, its use as a descriptor within the grasp taxonomy.

The F-GRASP Taxonomy

Feix *et al.* [Feix 15] presented a comprehensive framework for defining grasp poses from a survey of previous studies discussing human grasps translated to robotic agents. For a complete visualisation of the GRASP taxonomy, see Figure 2.1. To apply this framework to the set of grippers presented in Section 3.2, this chapter creates a fabric-specific version (the F-GRASP taxonomy) that outlines the grasps utilised in deformable manipulation (Figure 3.11). This set of grasp poses describes the array of configurations seen from grippers addressing fabric manipulation. The level of granularity presented in the original taxonomy is sufficient for deformable manipulation, and no further grasp definitions are

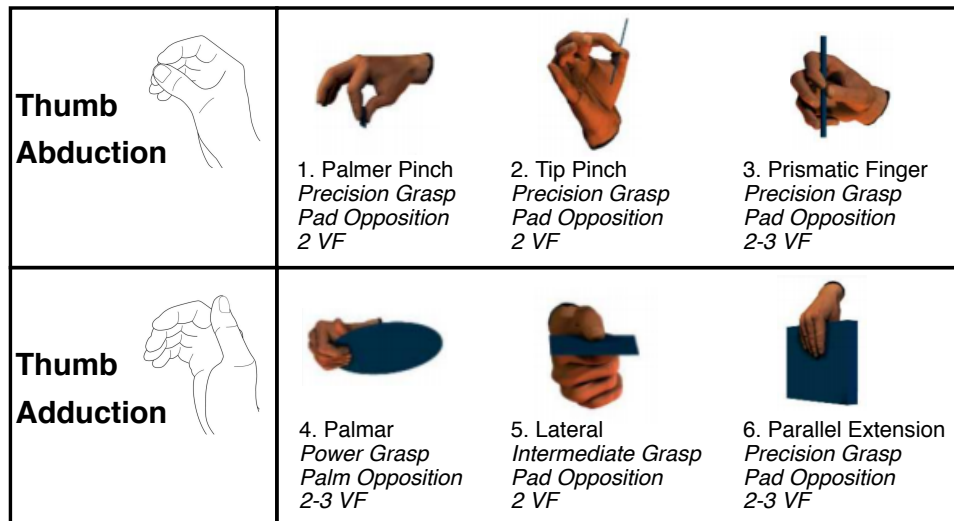


Figure 3.11: The F-GRASP taxonomy, an adapted form of the GRASP taxonomy, derived from Feix *et al.* [Feix 15].

required. An immediate observation is that the F-GRASP taxonomy requires only a limited range of anthropomorphic grasps. Figure 3.11 shows that only six grasps from the original 33 of the GRASP taxonomy were needed. Further details surrounding how these grasps apply to the grippers of Section 3.2 are present in Section 3.4.1.

3.3.2 Addressing Dexterous Manipulation

Defining Dexterous Manipulation of Fabric

The framework for classifying dexterous and in-hand manipulation was originally proposed by Bullock *et al.* [Bullock 11, Bullock 12]. A binary tree based on a series of parameters defines in-hand dexterous manipulation and contact descriptors surrounding a manipulation task. Actions are classified by the contact nature and motion parameters from a hand-centric viewpoint, enabling the definition of various robotic end-effector sequences of manipulation. Five parameters define a dexterous manipulation action as listed below. The abbreviations indicate whether the parameter is active and apply to Figure 3.14.

- Contact/No Contact (**C/NC**) – Refers to whether an interaction is making contact with an item or the environment.
- Prehensile/Non-Prehensile (**P/NP**) – An interaction with a single contact point represents non-prehensile interactions, while multiple contact points represent a prehensile action. Alternatively, if the grip can stabilise an object regardless of external forces, it is also considered prehensile.

- Motion/No Motion (M/NM) – Refers to an interaction where movement relative to a world coordinate frame occurs.
- Within-Hand/Not Within-Hand (W/NW) – If a motion is present, this parameter describes whether the movement comes from the wrist or occurs within the manipulator.
- Motion at Contact/No Motion at Contact (A/NA) – If contact and motion are present, this parameter details whether contact is significantly translating or rotating a held object concerning coordinate frames based on the contact points.

The resulting binary tree that constructs the IHDM taxonomy can express the complexity of the manipulation while describing the key aspects (Figure 3.12). While alternative taxonomies discuss grasps or specific manipulation primitives, Bullock *et al.* [Bullock 12] provide an abstract description of anthropomorphic behaviour without relying on object properties. While effective for a range of actions, Bullock

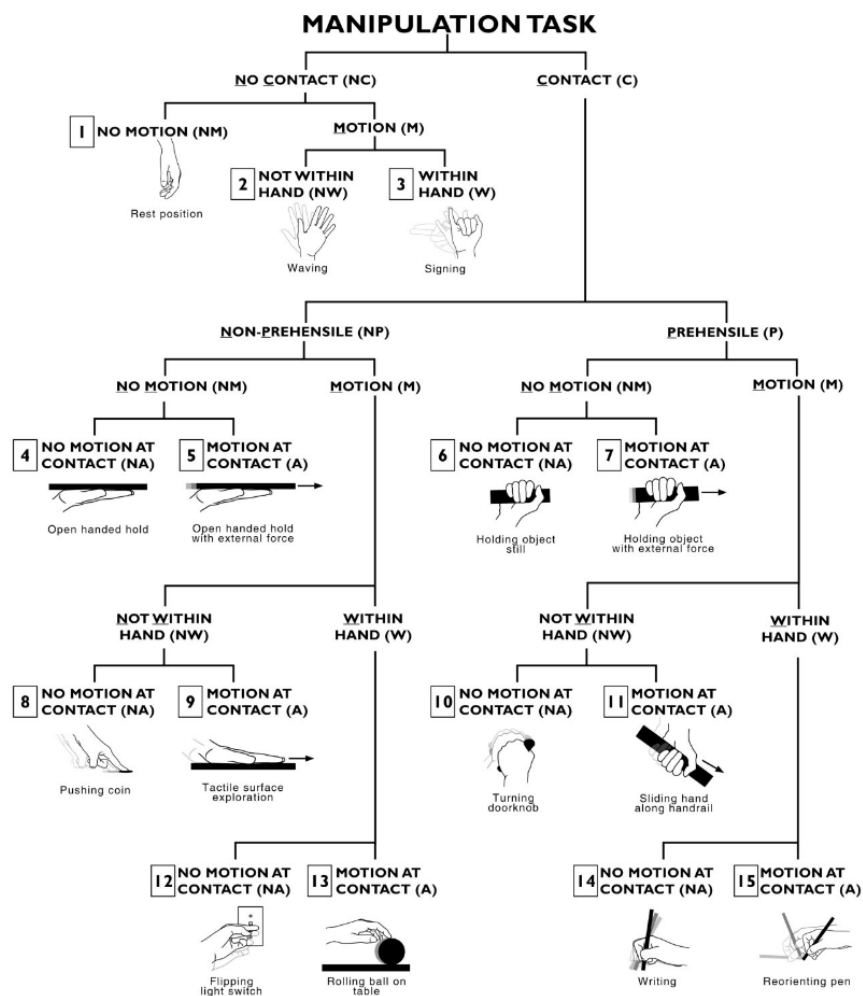


Figure 3.12: The original IHDM taxonomy of Bullock *et al.* [Bullock 12].

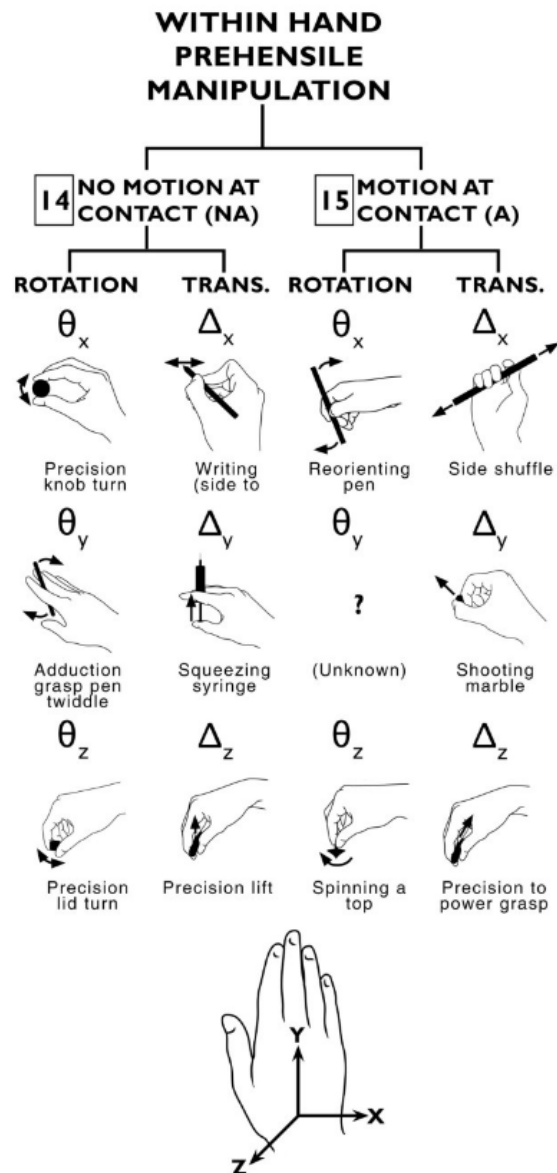


Figure 3.13: An expansion to scenarios **14** and **15** of the IHDM taxonomy (Figure 3.12), which describes the rotation and translation of within-hand manipulations.

et al. [Bullock 12] also acknowledge that more complex behaviour requires multiple labels, thus describing the sub-actions within a manipulation. An example given by Bullock and colleagues includes typing on the phone with a thumb. Such an interaction would involve a prehensile, no-motion manipulation (action **6** from Fig. 3.12) of holding the phone, combined with the non-prehensile/motion/within-hand task (action **12** from Fig. 3.12) of typing with the thumb. Additionally, the IHDM taxonomy provides an expansion classification scheme under scenarios **14** and **15** from Figure 3.12. These scenarios refer to situations where an object is translated or rotated during a within-hand manipulation. Figure 3.13 presents this expansion describing the direction of within-hand manipulations.

The F-IHDM Taxonomy

Similar to the way in which the F-GRASP taxonomy was defined, a significantly reduced version of the IHDM taxonomy called the F-IHDM taxonomy (see Figure 3.14) defines fabric manipulation actions specifically. This version uses components from both the initial binary tree (Figure 3.12) and the direction within-hand descriptors (Figure 3.13). Some actions from the initial binary tree are not present, as the subset of grippers did not perform these descriptive actions while manipulating fabric, resulting in a more compact representation. As an example, a stationary non-contact manipulation (scenario (1) in Figure 3.12) merely describes a stationary manipulator, not in use or waiting for conditions to operate. Thus, it does not add previously unknown information while describing fabric manipulation primitives. For simplicity, describing a gripper moving to a pose throughout a manipulation can be described under action (1) *NC-M-NW* per Figure 3.14. The classifications in the F-IHDM taxonomy can describe var-

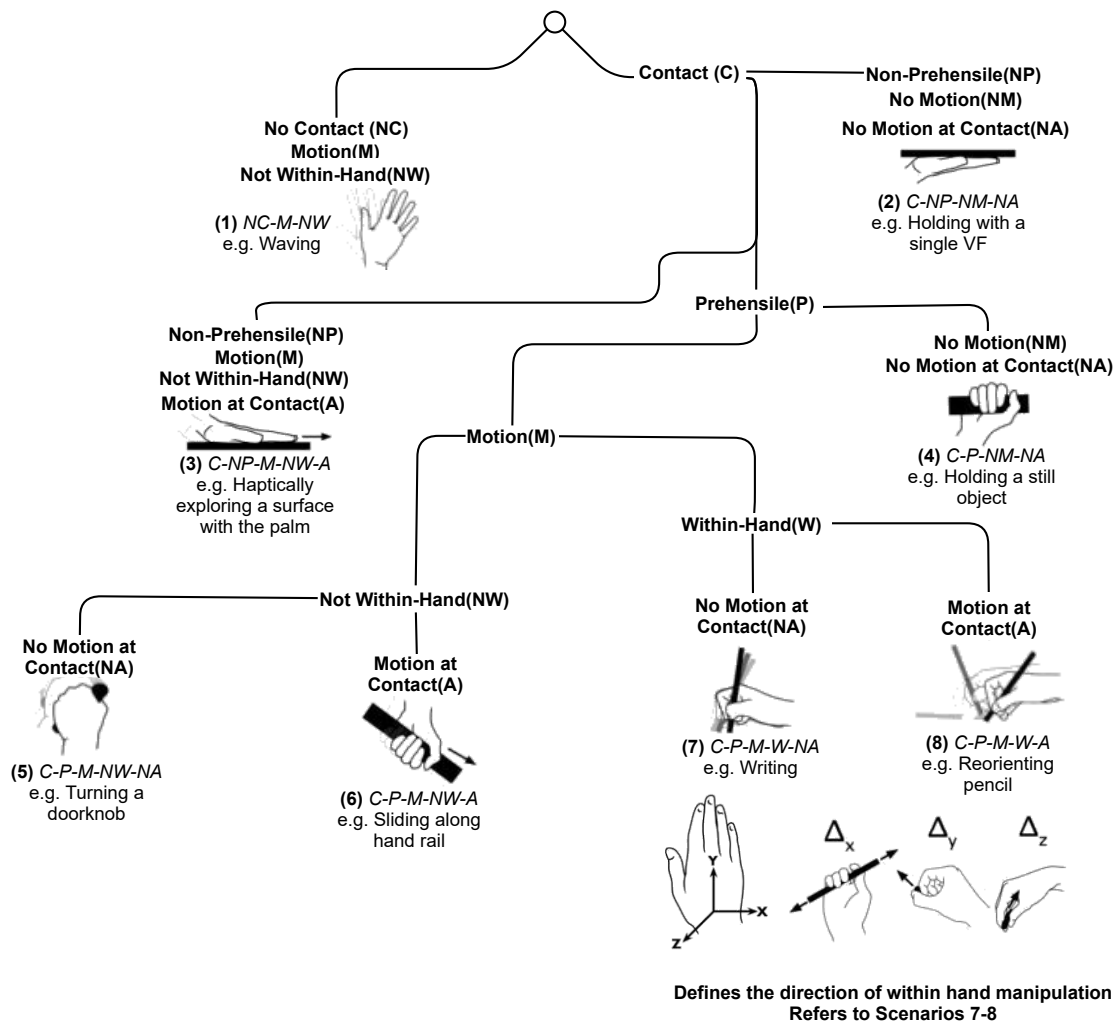


Figure 3.14: The F-IHDM, an adapted form of the IHDM taxonomy, derived from Bullock *et al.* [Bullock 12].

ious capabilities of the previously discussed gripper subset, including EC grasping, haptic exploration, flattening, grasping and grasp gaiting. Additionally, only the direction components from Figure 3.13 are used to describe motion direction for within-hand actions. Rotational descriptors are not included as fabric conforms to an applied grasp and held garment rotation will rarely apply. Section 3.4.2 discusses these skills with definitions from the F-IHDM taxonomy.

3.3.3 Application Towards Fabric Manipulation

These two taxonomies independently capture different aspects of anthropomorphic dexterous manipulation. As Bullock *et al.* [Bullock 12] suggest, combining the IHDM taxonomy with a grasp classifier can aptly describe grasp pose and behaviour throughout manipulation sequences. This suggestion drives the taxonomy discussion, observing and comparing how manipulators designed toward fabric manipulation approach their desired applications. The F-GRASP taxonomy, Figure 3.11, provides definitions that can be applied to a manipulator's pose while holding fabric. The F-IHDM taxonomy, Figure 3.14, provides a series of classifications to describe manipulator behaviour throughout a manipulation sequence. As a manipulator moves from a stationary position to grasp the fabric, assigned tags can formalise the stages of manipulation. For example, moving toward fabric would be classified as *(I) NC-M-NW* before grasping the fabric with a two-fingered pinch grasp using the *(8) C-P-M-W-A* action. After or during the manipulation, poses from the F-GRASP taxonomy can describe the manipulator's configuration while holding garments. Bullock *et al.* [Bullock 12] go on to suggest that one could observe correlations between grasp pose and manipulation behaviour. The desired outcome of this chapter is establishing a simple gripper design by understanding the grasp poses previous grippers use and their dexterous skills. The F-GRASP and F-IHDM taxonomies provide a framework to survey previous devices developed for deformable manipulation, discovering gaps in the present state-of-the-art. These gaps then inform the requirements to build a novel manipulator targeting fabric recycling applications.

3.4 A Treatise on Anthropomorphically Defined Grippers

This section presents a formalised anthropomorphic investigation of the surveyed grippers, beginning with the effector pose and the grasping forces seen across manipulators. Such a step uses definitions from the F-Grasp taxonomy (Figure 3.11). Next, descriptors from the F-IHDM taxonomy (Figure 3.14) describe dexterous manipulation behaviour observed throughout the investigated grippers. Alongside

these discussions, this chapter also describes the technical details and modelling methods used in previous devices' development. Finally, an overview of the various manipulators' capabilities and behaviour highlights critical takeaways and observations to consider when developing a novel robot manipulator.

3.4.1 Grasping

Grasp Poses Observed

When defining the actual grasp poses the evaluated manipulators adopted, one can see only a limited range of grasps, as shown in Figure 3.11. As some manipulators move away from an anthropomorphic form, other considerations inform the pose equivalent based on alternative characteristics. These features include fingertip contact regions, gripper motions and components' positioning. Some grasps defined in the taxonomy can have small but unique differences when applied to the manipulators, for example, the *Palmer Pinch (1)* and the *Tip Pinch (2)* poses (Figure 3.11). Under the F-GRASP taxonomy these grasps can be considered the same, as they remain a *Precision Grasp, Pad Opposition, 2VF* and *thumb abduction* grasp. While the differences remain minimal, one discrepancy is the region of the index fingertip in contact with the thumb. *Palmer Pinch (1)* will press the fingers symmetrically together while *Tip Pinch (2)* presses the finger and thumb together at an acute angle. Most of the discussed grippers utilise some precision grasp, partially due to some designs mimicking the thumb and index finger. Grippers **2** and **4** both use only a precision grasp with two virtual fingers. However, Gripper **2** exhibits fingertip contact characteristics similar to the *Palmer Pinch (1)* grasp while Gripper **4** more closely resembles the *Tip Pinch (2)* pose.

This distinction in fingertip contact also describes the difference in grasp pose taken between manipulators such as Gripper **1**, *Palmer Pinch (1)*, and Gripper **3**, *Tip Pinch (2)*. Gripper **1**, when grasping, tends to align the fingertips as they symmetrically press against each other. In contrast, Gripper **3** moves the fingertip into the thumb-like appendage at a pose consistent with *Tip Pinch (2)*. These grippers can also press two virtual fingers into the thumb or the held garment, capable of the *Prismatic Finger (3)* grasp. However, Gripper **1** uses the *Prismatic Finger (3)* grasp in a unique manner. While three virtual fingers were present, with the contact points similar to a *Prismatic Finger (3)* pose, the original authors [Donaire 20] proposed using this grasp more like a plane. The split lower fingers create a geometric plane that allows for specific manipulations, including folding clothing or holding folded garments without losing the folded configuration. While this grasp pose is analogous to the *Prismatic Finger (3)*, this section highlights how Gripper **1** uses this kinematic configuration in a unique method while manipulating fab-

ric. In Figure 3.15, the symbol \square denotes this grip creating a virtual plane, a reference originally used by Donaire *et al.* [Donaire 20].

The most diverse Gripper in terms of possible grasping configurations is the anthropomorphic three-fingered device, Gripper **6**, as also noted by Borràs *et al.* [Borràs 20]. The original development of Gripper **6** [Ono 01] details several specific configurations of the manipulator's grasping poses, which align to the grasp poses of *Tip Pinch (2)*, *Prismatic Finger (3)*, *Palmer (4)*, *Lateral (5)* and *Parallel Extension (6)*. The grippers **3** and **6** attempt to replicate the anthropomorphic structure of the human hand with a palm, thumb and multiple finger appendages. The other effectors used simple precision grasps of *Tip Pinch (2)* (Gripper **5**) and *Palmer Pinch (1)* (Gripper **7** and Gripper set **8**).

Interestingly both grippers **7** and **8.3** present a device where a rail controls the movement of two sub-clamp devices back and forth. While one could consider this grasp pose a unique single-hand grasping configuration, the limited applicability of both devices and the non-anthropomorphic nature of such a pose makes defining such an act both complex and redundant. Additionally, the inspiration for Gripper **7** came from a bimanual manipulation primitive. Thus the grasp assigned to these devices considers each sub-clamp a grasp in a bimanual manipulation rather than a single device. Figure 3.15 shows the discussed grasp poses alongside the grasp forces, which are addressed in the next section.

Grasp Forces

A wide range of grasping forces are present within the evaluated devices, but only a limited number of reports detail their effector's applied grasp force. For example, as described in Section 3.2.2, Gripper **2** used a human observation exercise to evaluate grip strength requirements and found their manipulator should be capable of a maximum grip strength of $\sim 30N$ to extract a garment weighing 500g with an extraction force of $\sim 35N$. These requirements derive from considering extracting clothing from an entangled state. Gripper **2** was adopted as an industrial-type device capable of a maximum grasping force of $40N$ [Le 13], and a reduced grasping force of up to $10N$ when performing haptic exploration on a garment [Denei 17, Le 15b]. The remaining grippers explicitly presented significantly reduced grasping forces between $0-5N$. Gripper **4** was capable of a maximum grasping force of about $4.7N$, while the devices in Gripper Set **8** were capable of a range of grasping forces up to approximately $5.5N$. However, these grippers were limited in scope towards in-hand-dexterous manipulation and holding scenarios. They did not explore the entangled garment extraction scenario Gripper **2** evaluated, indicating that a reduced grasping force may be applicable for most generalised cloth manipulation tasks.

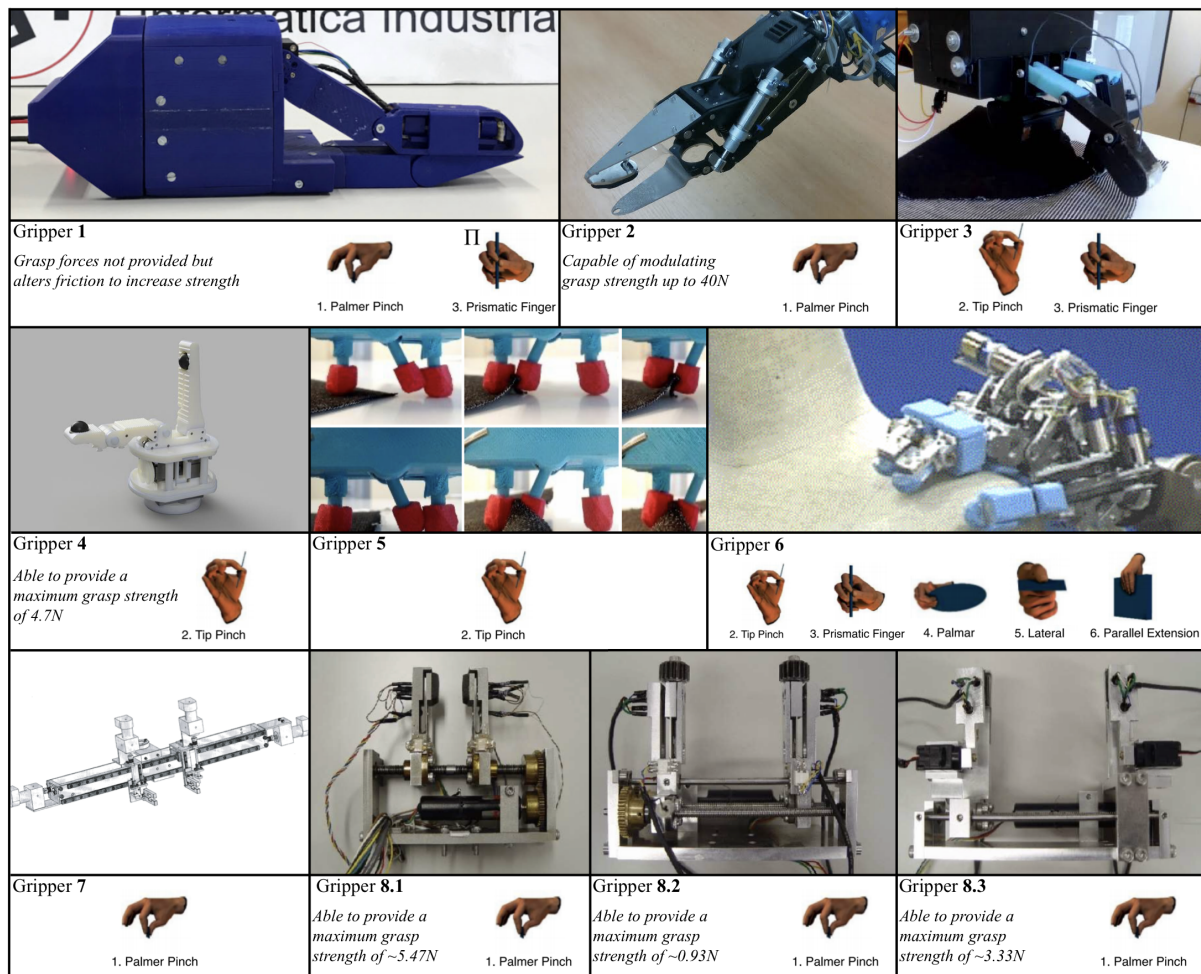


Figure 3.15: A summary of the grasping characteristics of the surveyed devices.

Grasp force is not the only parameter determining grip strength. Friction also plays a role. Several devices within the surveyed grippers present mechanisms to increase the friction between the held fabric and manipulator, thus increasing the grip strength. For example, Gripper 1 uses an embedded mechanism in the fingertip that extrudes and retracts a silicon pad, thus adjusting the friction between the effector and held fabric and changing the grip strength. As garments can be diverse in their physical and mechanical characteristics, it can be challenging to define precisely how this mechanism improves grip strength. The authors of Gripper 1 [Donaire 20] evaluate this device by observing the duration and success when holding fabric at various weights. Their manipulator held three different fabric items: a towel, a T-shirt and a silk item with additional weights ranging up to 250 grams, while the silicon pad was retracted and extruded. When the end-effector was in a high friction state (the pad extruded), stable grasping occurred with payloads up to a total weight of approximately 200g. However, when holding silk, a material with a lower friction coefficient, the device could only hold the fabric indefinitely at a weight of 70.5g. When

Gripper **1** retracts the silicon pad, the target application is a grasp gaiting act that allows the material to slide through the fingers. Similarly, Gripper **8.2** attempted to reduce the friction between the clothing and manipulator to allow for grasp gaiting behaviour. The original authors of Gripper **8.2** discussed how an inner finger was present to perform a robust grasp when a firmer grip was required.

From the surveyed devices, it appears that clothing manipulation is a task that, depending on the context of the manipulation primitive, can require a wide range of grasping forces up to $30N$. Many surveyed devices exhibit a limited grasping force of up to $5N$, possibly due to target applications that only consider manipulating a single item or light garments. Alternatively, hardware or design limitations could be responsible for the reduced grasping forces. The only device that exhibits a more significant grasping force is Gripper **2**, which specifically considered clothing in a tangled extraction scenario. However, this grip strength reduces to a maximum of $10N$ for haptic exploration tasks on a garment. Such an observation indicates that a manipulator targeting fabric manipulation should be capable of at least temporarily sustaining high grasping forces to handle entangled extraction scenarios or heavy garments. However, for most straightforward tasks, reduced grasping forces up to $10N$ should remain sufficient. Many devices also note the significant role that friction plays in grip strength, and make technical augmentations to adjust the friction while grasping. However, the mechanisms to alter the friction seen in the reviewed subset include complicated components such as roller fingertips or pad extrusion fingertips. Such elements add to the complexity of device design and can unpredictably interact with the various mechanical properties of different textiles, as exhibited by Donaire *et al.* [Donaire 20]. A more direct solution would be modulation of the grip strength as demonstrated by Le *et al.* [Le 13].

3.4.2 Dexterous Skills within Fabric Manipulation

Grasping clothing with environmental constraints

Fabric can present itself in states which make the initial grasping action difficult. As previously described, EC grasping allows humans to combat uncertainty and perform a robust grasping motion. While in a flattened state, fabric requires this collision-rich skill during manipulation. From the subset of grippers described in this chapter, one can observe two broad techniques of EC grasping applied. This research identifies those methods as biomimetic or insertion grasping, as visualised in Figure 3.16. Biomimetic grasping is an action that involves dragging a finger across flattened fabric, producing a protrusion in the fabric's body for the effector to grasp. Alternatively, an insertion approach applied by grippers **1** and **2** could also perform EC grasping by sliding a thin fingernail-like appendage between the



(a) An example of biomimetic grasping.

(b) An example of insertion grasping.

Figure 3.16: Examples of two grasping techniques that exploit the environment.

fabric and environment surface, followed by a clamp grasp on either side of the fabric's body.

These methods offer differing benefits and suffer from distinct limitations. Before elaborating on these aspects, this section defines the pipeline of manipulation for these techniques via definitions from the F-IHDM taxonomy (Figure 3.14). To visualise the process of a biomimetic grasping act, Figure 3.17 presents an image with Gripper **3** performing a biomimetic grasp on a towel using definitions from both taxonomies as previously derived. Note that the tag **exC** denotes situations where environment surface contact is taking place.

In Figure 3.17, biomimetic grasping occurs in a situation where all fingers are in contact with the towel, and use the environment to deform the material into a grasp. Thus contact points are constantly changing between the fabric and environment surface. However, little to no motion occurs at the contacts between the manipulator and the fabric as grasping occurs. In this example, the visualisation uses the F-IHDM classifications referring to contact between the manipulator and the fabric only. It should be noted that biomimetic grasping can occur in situations with different contact parameters, for example, where the EC grasp starts at an edge, colliding with the environment surface before interacting with fabric during the EC grasping motion. From Figure 3.17, one can observe that when deforming the material, a classification of **(8) C-P-N-W-A** is applied to highlight the minimal motion at contact between the gripper and fabric while EC grasping.

Alternatively, insertion grasping preserves the initial state of the garment by performing a motion that constantly shifts the contact between the manipulator and the fabric. Figure 3.17 highlights this process by showing Gripper **1** grasping a folded T-shirt while mostly retaining the folded configuration. Unlike

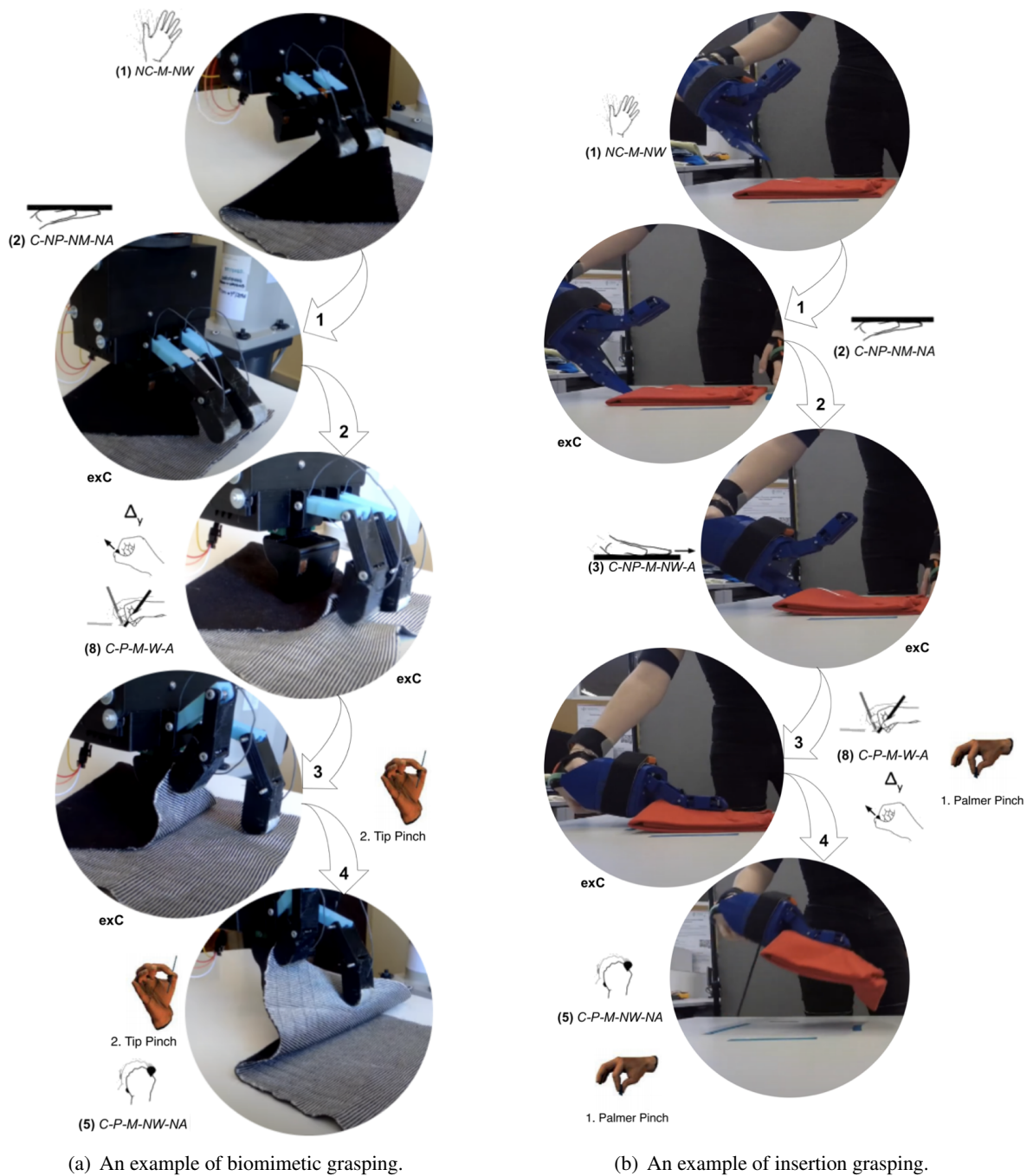


Figure 3.17: Grippers 3 (left) and 1 (right) performing EC grasping with dexterous manipulation annotations and grasp pose tags.

biomimetic grasping, within-hand motion does not occur while executing insertion grasping. Instead, a non-prehensile sliding action inserts a thin finger between the fabric and the environment surface before closing the grasp, as shown in Figure 3.17. Both devices which exhibited this behaviour (grippers **1** and **2**), used a thin finger which slid between the garment and workspace surface with compliant mechanical components, allowing the device to collide with the environment safely.

Insertion grasping is intrinsically limited when one considers that it can only occur where the edge of a garment meets the table. While this factor limits the suitable grasping locations and can require compliant components, the core benefit of such a technique remains the minimal disturbance to the target garment's state while executing this action. While developing Gripper **1**, Donaire *et al.* [Donaire 20] show this method grasping in multiple scenarios, including a garment on a surface and extracting a single item from the top of a folded pile of laundry. Additionally, both manipulators capable of insertion grasping also reference the skill of 'flattening', a motion involving a single VF sliding across a flat garment, removing wrinkles and spreading the material. Under the F-IHDM taxonomy, such a motion falls under classification **(3) C-NP-M-NW-A**.

One can consider biomimetic grasping to have an inverse set of limitations compared to insertion grasping, as biomimetic grasping relies on deforming the material to grasp. However, such a property means grasping can occur anywhere on the fabric's body, including on the edge of a garment. Thus the fingertip will drag along the environment surface before catching the fabric to grasp. While there are variations of biomimetic grasping, including the hand in contact with the surface before grasping, the direction of the virtual finger relative to the palm, the number of fingers moving within the interaction and the final grasp pose, the various techniques remain quite similar. A contact, generally a non-prehensile interaction, is made against the surface, then one or more appendages move inwards to produce a protrusion, which can be clamped upon by the effector. Half of the manipulators from the surveyed subset were able to perform some kind of biomimetic grasping motion, including grippers **3**, **5**, **6** and **7**; nevertheless, across these devices this review notes several common limitations.

Firstly, biomimetic grasping generally uses position controllers that move the fingertips to brush against the surface in pre-programmed trajectories. Such an approach did not result in robust grasping behaviour for various reasons; Koustoumpardis *et al.* [Koustoumpardis 14] noted an issue where Gripper **3** fails to grasp fabrics with a low friction factor while performing biomimetic grasping. To alleviate this issue, the authors propose changing the fingertips to use a silicon material with a higher friction factor. Gripper **7** also uses silicon pads to increase the friction between the fabric and manipulator while

performing biomimetic grasping.

Adding silicon materials to fingertips does increase the friction coefficient between the gripper and fabric, improving the success of biomimetic grasping motions with low-friction fabrics. However, this solution would not generalise well; increasing this friction would require the manipulator to exert greater forces while navigating interactions with a higher friction coefficient. Consider the scenario where a gripper has to drag along a table surface before interacting with fabric while grasping. While dragging along the table, more actuator torque would be required to overcome the increased friction coefficient. A more elegant solution to low-friction interactions would be a form of actuator force control, thereby modulating the wrench exerted at the fingertips. However, none of the manipulators capable of biomimetic grasping can control the force exerted on the environment surface while dragging inwards. The complexities of friction and normal forces applied while performing biomimetic grasping were also explored by Ono *et al.* [Ono 01] while developing Gripper 6 when extracting flattened garments piled on top of each other. Gripper 7 would also use biomimetic grasping to produce a stable protrusion rather than simply grasping deformed material as the other devices did. It would then release the material and re-grasp at the edge of the garment, a process visualised in Figure 3.18. However, such a solution would not generalise to a wide range of fabrics due to the varying non-linear mechanical behaviours.

The noted last issue about biomimetic grasping was the inability of devices to execute this behaviour from an arbitrary range of wrist orientations. Instead, the grippers usually approach the flattened fabric

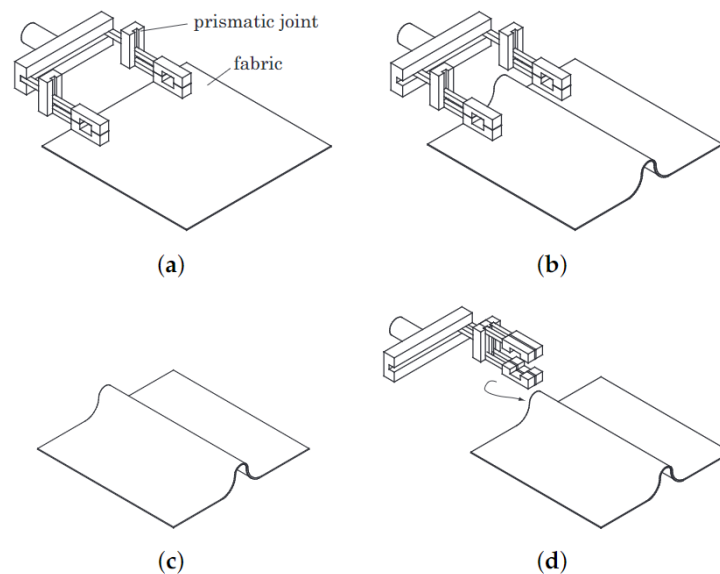


Figure 3.18: Gripper 7 performing a unique biomimetic grasping technique, sourced from Shibata *et al.* [Shibata 16].

from a specific wrist orientation where the palm was parallel to the fabric, as seen by all the grippers performing biomimetic grasping. While this notion was true for flat environment surfaces, Gripper **6** also performed biomimetic grasping on ‘non-flat’ environment surfaces such as a fashion mannequin’s shoulder, which required unique wrist orientations. Additionally, Gripper **3** was capable of sliding fabric to the edge of the table before moving the palm to a perpendicular orientation to the surface and grasping [Koustoumpardis 14]. Despite these limited capabilities, the skill of being able to execute biomimetic grasping from a range of wrist orientations on a flat environment surface, henceforth called *arbitrary* grasping, remains one which the surveyed devices were incapable of utilising.

Another observation relevant to robotic EC grasping is that any manipulator capable of insertion grasping is not capable of biomimetic grasping. This can be due to one or more factors. Firstly, the manipulators targeting insertion grasping have specifically designed fingertips with thin geometric proportions to slide between limited regions to execute insertion grasping. In contrast, grippers targeting biomimetic grasping focus on dexterous in-hand skills to execute the grasp. Additionally, insertion grasping uses motions provided by the arm appendage to execute the grasping motion.

In-hand dexterous skills applied to fabric manipulation

Once holding fabric, acts of dexterous in-hand manipulation such as haptic exploration and grasp gaiting are often observed among the grippers. This section details how these skills are applied and identifies their further applications. Inferring tactile information via haptic exploration has proven to be a valuable capability on effectors targeting deformable manipulation. Haptic exploration regarding fabric is an in-hand motion where two virtual fingers perform a ‘rubbing’ motion to infer the fabric’s features. Applications include the classification of held garments, detecting slippage or determining grasp success. Two manipulators from the discussed subset were capable of this skill, grippers **2** and **4**. Gripper **2** contained an embedded multi-modal tactile sensor for both classification and slippage detection, whereas Gripper **4** placed triaxial force sensors at the fingertips to complete the same set of tasks. Qualifying the haptic skills of these two effectors is relatively simple. As mentioned when discussing grasp poses, Gripper **4** holds a precision grasp that could be considered a *Tip Pinch (2)* grasp while Gripper **2** uses a *Palmer Pinch (1)* configuration. However, when described by the anthropomorphic framework, both effectors’ resulting in-hand dexterous behaviour is identical. A translation moves the contact points bidirectionally along the z axis of the coordinate frame defined in Figure 3.14. One could describe this motion as haptic exploration perpendicular to the palm. It can be complex to describe this motion using the F-IHDM

taxonomy, but for the sake of simplicity, this chapter assumes that contact points change between the finger and cloth while rubbing the material. Thus it comes under scenario **(8) C-P-M-W-A**.

Several grippers also capable of grasp gaiting, where the grip of a garment is never released, but the grasp point is adjusted. Many of the manipulators discussed explicitly demonstrate or target this skill, including the grippers **1, 4, 7, 8.1, 8.2** and **8.3** (Figure 3.19). One simple solution to perform grasp gaiting is to apply a minimal grasp force that allows the fabric to slide between the held grip. Such a skill is heavily connected to grasp force modulation capabilities and the majority of these devices take this grasp force adjustment approach. Thus under F-IHDM definitions, grasp gaiting in this manner fits scenario **(6) C-P-M-NW-A**. Gripper **1** performs grasp gaiting by sliding across the material with the friction alteration mechanism retracted in the fingertip to allow the fabric to slide between the fingers. Gripper **4**'s description briefly details grasp gaiting behaviour, and associated media files highlight this gripper performing grasp gaiting acts.

Devices **7, 8.1** and **8.2** apply an appropriate grasp force to allow the material to slide between the fingers; it is also noted that Gripper **2** should also be capable of grasp gaiting in this manner using its grasp force modulation capabilities. However, grasp gaiting behaviour was not presented in associated media files nor publications describing Gripper **2**. Finally, the only device to perform grasp gaiting using an in-hand manipulation was Gripper **8.3**. As described in Section 3.2.8, this device can perform an in-hand manipulation by using two sub-clamps which release and regrasp fabric while moving along a prismatic rail. Given the assumption is made that fabric is constantly sliding across the open sub-clamp, such a motion could meet the manipulation classification of **(8) C-P-M-W-A** per Figure 3.14. Details about the dexterous skills of these manipulators are also presented in Figure 3.19.

3.4.3 Applications

So far, this discussion covers the dexterous skills and methods of approaching EC grasping these various manipulators apply. This subsection describes each manipulator's applications to put these skills in context. To start, some manipulators target individual dexterous skills as part of larger manipulation tasks. For example, Grippers **3, 5** and **7** simply target grasping flattened fabric from the environment. While such an act is part of a broader manipulation pipeline, i.e. unfolding or spreading fabric, the unique skill of these manipulators is in performing a form of biomimetic grasping to grasp flattened material. Interestingly, the devices that used insertion grasping to navigate the environment, Grippers **1** and **2**, included folding in their applications. Such an observation is consistent with the notion that insertion





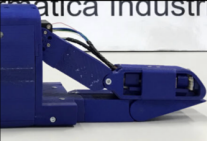

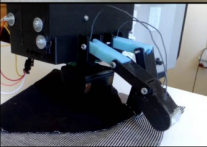

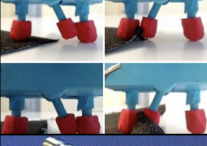
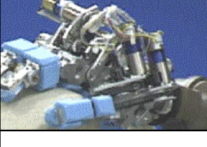

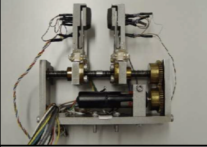
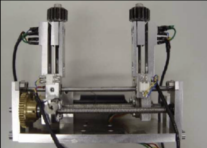
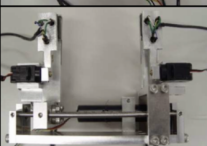
	 <i>Insertion Grasping</i>	 <i>Biomimetic Grasping</i>	 <i>Haptic Exploration</i>	 <i>Grasp Gaiting</i>
<i>Gripper 1</i> 	✓			✓
<i>Gripper 2</i> 	✓		✓	✓ (dashed)
<i>Gripper 3</i> 		✓		
<i>Gripper 4</i> 			✓	✓
<i>Gripper 5</i> 		✓		
<i>Gripper 6</i> 		✓		
<i>Gripper 7</i> 		✓		✓
<i>Gripper 8.1</i> 				✓
<i>Gripper 8.2</i> 				✓
<i>Gripper 8.3</i> 				✓

Figure 3.19: A summary of the dexterous skills from the discussed devices.³

³The skill of grasp gaiting is present under *Gripper 2* with a dashed line as the literature did not indicate that this skill explored by the authors. However, the grasp force adjustment abilities imply that such skill should be possible on this platform.

grasping retains a garment's initial state. Thus, any device targeting folding-based tasks requires the skill of insertion grasping. Gripper **1** specifically targeted the tasks of moving folded garments, folding a shirt, and tracing an edge, alongside generic manipulation skills such as insertion grasping, flattening clothing and holding fabric. Alternatively, Gripper **2** targets garment picking (accounting for entangled conditions), folding and general tasks such as sorting and unfolding.

Gripper **2** retained the broadest range of grasp forces ($0-40N$), which assisted when grasping entangled garments. However, this manipulator also performed haptic exploration at a reduced grasping force of $10N$, while the other devices had grasp forces up to $5N$. Furthermore, Grippers **1** and **2** remain the most ambitious regarding applications, yet both remain limited in terms of possible grasp poses. Both were capable of the *Palmer Pinch (1)* grasp configuration, and Gripper **1** was additionally capable of a *Prismatic Finger (3)* by splitting the lower finger appendage. However, Gripper **6** was capable of five grasp poses identified in the derived grasp taxonomy. Borràs *et al.* [Borràs 20] also note how Gripper **6** remains diverse in terms of possible grasp configurations; nevertheless, the application of Gripper **6** was limited to executing biomimetic grasping on environmental surfaces such as stacks of fabric, table surfaces or fashion mannequins.

Gripper **4** was not capable of EC grasping and only used the *Tip Pinch (2)* grasp. However, the desired applications were unique throughout the observed set of manipulators. The applications outlined by Von Drigalski *et al.* [Von Drigalski 17b] included bed-making or other tucking tasks. The skills required for such applications included grasping and sliding to pull a bed sheet taut while using arm motions and the static thumb appendage for tucking sheets into the bed. Finally, some manipulators only target the application of edge tracing, i.e. Gripper set **8**. Thus the only skill sought in these devices was grasp gaiting. Observing the applications of these manipulators highlights several key points for consideration. Insertion grasping is the preferred EC grasping method if folding is considered a necessary manipulation primitive. Next, a two-fingered precision grasp should remain sufficient for many fabric-based robotic applications. In addition, any generalised solution should possess grasp force modulation capabilities up to a maximum grasp strength of $30N$ as outlined by Le *et al.* [Le 13].

3.4.4 Technical Review of Surveyed Grippers

This subsection addresses the technical details of the grippers considered, including the compliant components, modelling approaches, the number of actuators present in a system, and system controllers. The

devices capable of insertion grasping, Grippers **1** and **2**, used compliant elements in the fingers that collided with the environment. Gripper **1** defines their compliant joints in the lower fingers with a flexible metal plate in a rotational joint limited to 0.523599 radians of motion, although no modelling information about compliance is presented [Donaire 20]. Gripper **2** took a different approach where variable impedance actuators (VIAs) control the compliance of certain joints at different stages of manipulation to safely collide with the environment. A kinematic chain defining the manipulator and subsequent steps are presented in two articles by Le *et al.* [Le 13, Le 15b], which allows the authors to use classical mechanics to create a Kinematic and Kinostatic model. Kinostatic modelling took place using planar twists and wrenches, and screw theory solved the velocity kinematics. This process also involved modelling the VIA itself, allowing the developers to comprehensively understand their device and its associated grasping capabilities. The final design of Gripper **2** is a planar device with nine defined joints; two linear actuators drive each finger in a series-parallel mechanism, and the VIA controls compliance at the finger appendage.

Some descriptions do not provide a model of the gripper or the underlying control mechanisms. An example is the originating publication of Gripper Set **8** [Sahari 10], which only presents the manipulator components and design aspects without demonstrating a modelling procedure. However, when developing Gripper Set **8**, the focus was not on developing new manipulator designs but on observing simple devices and evaluating their edge-tracing capabilities. In addition, for some grippers, limited kinematic modelling is presented. Koustoumpardis *et al.* [Koustoumpardis 14] provide a tendon-driven kinematic and a kinostatic model while developing Gripper **3**. However, this modelling did not consider scenarios where the environment was present and estimated the forces needed to move and hold grasp poses. The description of Gripper **5** presents the geometric equations to drive the CAM-follower mechanism, but further modelling is limited. Von Drigalski *et al.* [Von Drigalski 17b, Von Drigalski 17a] did not present modelling procedures when describing Gripper **4**. However, the inspiring publication [Ma 16] provides a system model that maps the tendon forces, joint forces and fingertip wrenches. Finally, the articles describing Grippers **6** and **7** do not provide a modelling process for their manipulators. However, Ono *et al.* [Ono 01] present a kinematic diagram of Gripper **6**.

Another aspect to consider is the DoF and DoA (degrees of actuation) of the observed devices. As the DoF or DoA of a device increases, the development and modelling procedures become more complex. Ideally, the design of any novel device developed for fabric manipulation should try to minimise the DoF and DoA parameters while maximising grasping and in-hand manipulations. Several devices in

the surveyed set are underactuated, meaning they have more DoF than DoA. For example, Gripper **3** only used two rotational actuators embedded in the palm, each controlling a finger appendage with three joints via tendons. Additionally, a passive thumb appendage was present to evaluate collisions. Gripper **5** used a single actuator to control a shaft with CAM mechanisms to move three finger appendages. While both devices **3** and **5** could perform EC grasping, they lack further capabilities. Gripper **8.2** could also be considered underactuated as a single actuator operates the device in a simple parallel gripper structure, but roller fingertips rotate freely on bearings to improve edge tracing. One could also consider the devices with compliant joints underactuated, i.e. Grippers **1** and **2**. Interestingly, Gripper **4** has a unique ability to switch between modes of under-actuation. Ma *et al.* [Ma 16] introduce this capability and describe the tendon routing and modelling approach that allows this device to execute fully actuated and underactuated motions. This device uses two actuators with tendon routing through a planar finger with two joints.

Gripper **6** has seven joints and is also underactuated, as the final two joints on the finger serial-chains are subserviently linked to the previous actuators. The remaining devices used actuators to control each DoF. For example, Gripper **8.1** operates as a simple parallel manipulator, while Grippers **7** and **8.3** are similar in structure, with two clamps operating along a rail. However, Gripper **8.3** uses three actuators and can only move a single clamp device along the prismatic rail, whereas Gripper **7** is physically larger and can move both clamps along the rail using four actuators in total.

Finally, the actuator control mechanisms are not detailed for most devices surveyed. This aspect can influence manipulator behaviour, including dexterous motions and grasp forces. Some devices, including Gripper Set **8**, provide information about the actuators and encoders used in their setup but lack control details. Similarly, for Grippers **3**, **4**, **5**, **6** and **7**, some technical aspects are detailed, such as the actuators used, but lack information about a specific form of control. For these devices, one can assume that a form of position control was present as pre-programmed trajectories or poses usually informed the manipulator operation. Experiments with Gripper **1** used position control, but the authors [Donaire 20] mention how the actuators are capable of torque control. The device with the most comprehensive modelling process in its description, Gripper **2**, also has a control system for the linear actuators and VIA implemented [Le 15b]. This system assists the gripper in modulating the grasping strength throughout the required tasks.

3.4.5 Open Areas to Address

The discussion so far has covered grip strength, hand pose, and dexterous behaviour that research-developed grippers use to manipulate fabric. The diverse range of grasping poses of Gripper 6 [Ono 01], had limited use cases indicating that a wide range of grasp poses are not required for manipulators specifically targeting deformable manipulation. Especially when considering the most diverse grippers in terms of application, devices 1 and 2 only used precision grasps, reinforcing the perspective of point-to-point grasping being sufficient for a wide range of fabric manipulation tasks.

Biomimetic grasping could be considered the more generalised solution to fabric EC grasping when prioritising the initial grasp task. Such an observation aligns with the increased frequency of biomimetic grasping compared to insertion grasping displayed by the surveyed manipulators. Additionally, biomimetic grasping can apply to a more comprehensive range of applicable grasp points compared to insertion grasping, which can only occur from the edge of a garment. However, when considering the effectors' applications, the capability of insertion grasping would be necessary in any manipulator targeted toward assistive or domestic use cases, as this method does not significantly change the state of the cloth as a grasping action takes place, unlike biomimetic grasping.

None of the surveyed effectors could perform biomimetic grasping from various wrist orientations. Generally, this action took place from an angle where the palm was parallel to the surface where the grasp action would occur. No manipulator explicitly detailed a force control method while performing biomimetic grasping. It appears that the primary method was a heuristic approach, using pre-programmed trajectories to drag inwards. Manipulators that could perform biomimetic grasping were also incapable of haptic exploration. When observing the skill of haptic exploration, the examined effectors' dexterous actions would solely traverse in a linear direction perpendicular to the palm. Tactile exploration did not involve bidirectional motions, usually occurring linearly between fingertips.

Regarding grasp gaiting, the most utilised technique could be considered the most human-like approach, wherein slippage of fabric through the fingertips was allowed by dictating the applied grasp force. From a development viewpoint, this indicates that a constructed manipulator targeting deformable manipulation would require a control mechanism that could moderate the torque exerted by the grasping actuators within the system. Such a feature could also be beneficial if haptic exploration were integrated into the gripper design, as in Gripper 2, where the modulated grasp force could adjust to the task context.

Given these observations, one can observe various technical and functional limitations in the surveyed grippers. The design thinking process now performs the *ideate* step, formulating a gripper design

that can improve the existing state-of-the-art. The requirements of such a device include *arbitrary* grasping (biomimetic grasping from a range of wrist angles), grasp force modulation, haptic exploration, and grasp gaiting. These skills fulfil the requirements of a dexterous manipulator that can perform the pick-and-place sorting of textile waste. In addition to these desired capabilities, any proposed solution should remain a simple system with limited actuators and an established modelling process. To build a device that meets the technical criteria while retaining the desired skill set, actuators that form grasp poses and interact with the environment will require torque or impedance control aspects to modulate the grasp force and provide elements of control over the wrench exerted at the fingertip. The actuators using impedance or force control must also be back-drivable and capable of the required stall torque parameters for grasp strength up to approximately $30N$. In addition, developing a novel solution should avoid complex features such as tendon-driven designs or passive compliant joints.

Previous devices have used human morphology as an inspiration for design, for example, mimicking the thumb and two fingers (grippers **1**, **3**, and **6**) or simply a thumb-finger pinch (grippers **2** or **4**). However, these approaches must make concessions when replicating features of the human hand, such as complex design features or size accommodations. In addition, replicating the human hand's complete motion can require many actuators, as shown by Xu and Todorov [Xu 16]. Considering these factors, this chapter proposes an alternative approach to gripper design. Thus, rather than broadly mimicking the morphological aspects of the human hand or designing a manipulator from a targeted set of manipulation primitives, a single grasp configuration from the F-GRASP taxonomy is used to inspire a design. The grasp inspiration must be capable of the skills described above, while simultaneously using a minimal number of actuators and maintaining a simple structure.

Therefore, this chapter proposes a gripper centred around the lateral grasp (see Figure 3.11) for simple pick-and-place applications with fabric. The lateral grasp is a grip where the thumb performs the majority of motions. Thus, a single serial-chain structure can represent actuation and needs only to replicate thumb motions. Additionally, actuation can occur at each joint and does not require subservient joints or complex actuation mechanisms. Formed grasps occur by pressing the thumb into the side of the index finger in a point-to-point grasping configuration. While this grip may be ill-suited to more refined deformable manipulation tasks such as folding or placing an item on a coat hanger, it remains an acceptable grasping configuration for pick-and-place sorting. Finally, pursuing a design inspired by the lateral grasp remains an avenue that previous research has not investigated. While the lateral grasp initially seems acceptable, considering the arguments above, further investigation is required. Therefore,

the final component of this chapter's research presents a user study that examines how humans respond to kinematic constraints while manipulating fabric to validate that such a conceptual design is sufficient for generalised textile waste sorting.

3.5 Human-Centric User Study

3.5.1 Introduction

This chapter has discussed several capabilities required to implement a more versatile gripper solution and proposed a conceptual design from an anthropomorphic hand-centric perspective. However, this discussion primarily addressed manipulation from a single-hand context. Thus, to establish a broader understanding of fabric manipulation and validate whether the proposed lateral grasp could be sufficient for a simple, generalised fabric manipulation effector, a complementary user study⁴ investigates humans manipulating fabric under various grasp constraints. Observations of human manipulation behaviour to inform robot operation within deformable manipulation are present in the literature, as for example, the video dataset of Verleysen *et al.* [Verleysen 20], who observe human folding of garments for transfer to robot behaviour. As the manipulator survey and broader literature indicate that precision or 'point-to-point' grasps should remain sufficient for simple manipulation tasks, the focus of this user study is to investigate whether the type of precision grasp impacts task ability or efficiency.

While dataset-based approaches towards deformable manipulation strategies are a viable research avenue, such endeavours usually involve gathering and annotating high-level motion planning methods involving computer vision and garment state estimation (e.g. Verleysen *et al.* [Verleysen 20]). A user study remains the most appropriate approach to validate the observations from Section 3.4 and investigate the impact of grasp constraints on deformable manipulation. In the future, when further investigating grasping strategies, a combined user study and data collection could reveal more refined aspects of deformable manipulation.

Investigating the grasp-type aspect of manipulation addresses two avenues through observing human behaviour. Firstly, the task duration for various clothing manipulation actions, such as folding and sorting, are studied. As Borràs *et al.* [Borràs 20] and Donaire *et al.* [Donaire 20] note, folding fabric could see improvement from grasp configurations beyond 'point-to-point' poses. As this experiment constrains the dexterous skills of humans by limiting their grasp configurations, one expected outcome is to observe

⁴The Human Research Ethics Committee (HREC) provided approval to conduct this user study as per University of Canberra policies and guidelines: 6982 - Humans and Cloth Manipulation: Deriving inspirations for robotic manipulation.

that folding tasks will take longer when constraining participant's grasping behaviour. As this user study seeks to validate that the lateral grasp will be sufficient for pick-and-place waste sorting, part of the experiment also includes sorting garments into various categories. If the lateral grasp is sufficient for simple grasping and sorting, then one would expect little difference in the duration of the sorting tasks as a consequence of the constraints. Finally, a post-observation questionnaire asks participants how they viewed the difficulty of manipulation while constrained. Such feedback can inform which tasks were made more difficult and to what extent when constrained. This user study attempts to understand how grasping constraints affect deformable manipulation by examining these aspects and validating that the lateral grasp is a suitable inspiration for a more refined grasping solution.

3.5.2 Method

In total, 20 participants performed various fabric manipulation tasks in this user study. The instructions provided to the participants were as follows. Firstly, the exercise began with a single T-shirt on a table in a flattened state with instructions for participants to grasp the shirt from the centre and edge several times. Then, following the grasping actions, instructions were given to the participants to fold and unfold the shirt three times. Once dexterous actions with the shirt were complete, participants moved to an unsorted heap of clothing made up of a necktie, scarf, two T-shirts, three long-sleeved collared shirts, one pair of trousers and two pairs of shorts. Participants then extracted a tangled item from the heap five times, defined as a garment item in contact with the table at the bottom of the heap. Participants then sorted the clothing by type, followed by a repetition of the sorting task but with folding.

Subjects repeated this list of tasks three times. As repetition occurred, constraints informed participants of the types of grasps they were allowed to use. Such an experiment uses a within-subject setup, where participants experience all constraints while under observation. Initially, participants are permitted to complete the tasks however they see fit. Then, upon repeating the tasks for the second time, instructions informed participants to use a pincer/pinch grasp, i.e. the thumb and index finger⁵. Finally, upon the third round of performing tasks, the constraint was use of the lateral grasp, in which participants could only hold clothing with their thumb and the side of the index finger. Incidental contact during manipulation was allowed. A four-camera setup in the University of Canberra HCT (Human-Centred Technology) observation lab recorded participants to capture audio and visual data streams, as visualised in Figure 3.20. The software Jamovi was used to conduct all statistical analyses in this chapter [Şahin 19].

⁵In this human-centric user study, the terms 'pincer' and 'pinch' are used interchangeably when referring to the thumb-index-finger constraint.



Figure 3.20: An example of the observation setup.

As previously mentioned, validation of the lateral grasp and observation of bi-manual manipulation skills are the desired outcomes of this user study. Thus, a data collection process takes place, which records the duration in seconds for participants to complete the tasks under various constraints from the audio-visual recordings. Upon collecting the data, an initial calculation of the three common measurements of central tendency occurs, followed by an in-depth statistical analysis. For the purposes of the analysis, the null hypothesis was: *There is no difference regarding temporal efficiency between constraints while manipulating fabric in various tasks.* Duration in seconds is a continuous dependent variable (DV). The independent variable (IV) is the applied hand constraint, which is categorical with three possible options (unconstrained, pinch and lateral). Finally, as each participant performed the tasks under all three constraints, the design of this analysis is structured as a within-subject study. Under these conditions, a compatible statistical evaluation is the repeated measures analysis of variance (RM-ANOVA). The data collected for this user study was on a limited number of participants, and the Jamovi software determined that the data was non-parametric. Therefore, the statistical interpretation of manipulation duration uses the non-parametric Friedman test alongside pairwise comparisons with the Durbin-Conover evaluation [Herath 22].

Upon finishing the outlined tasks, participants filled out a brief questionnaire asking how the grasp limitations impacted the perceived difficulty (Appendix A.2). Participants provided a score by placing

a mark on a line indicating from -100 (significantly easier) to 100 (significantly harder) for the tasks of grasping, folding and unfolding under each constraint as compared to the unconstrained scenario. The null hypothesis in evaluating this data was: *The constraints did not impact task difficulty compared to the unconstrained condition across manipulation tasks.* Interpreting this data follows a slightly different procedure than for the duration interpretation discussed above. Unlike the previous analysis, which contains three categorical variables, this analysis only uses the pinch and lateral constraints, which are both compared to the unconstrained condition. Aside from the two categorical variables, the other parameters match the duration analysis. Thus an appropriate test to perform is the Paired Samples T-Test [Herath 22]. Additionally, Jamovi executes the non-parametric alternative Wilcoxon Rank Test [Herath 22]. This evaluation also collects the three common measurements of central tendency for the difficulty interpretation data.

3.5.3 Results

The first observation from this study was that regardless of grasp constraints, all human participants could fold, sort, and manipulate clothing. Grasping from the environment and from an unsorted heap were both obtainable goals, along with folding and sorting tasks. This exercise also revealed examples of adaptation strategies while manipulating that the taxonomy discussion did not consider. These methods became apparent as constraints appeared across the experiment. For example, a participant's grasp strength or arm configuration may have been insufficient to extract the garment from the unsorted heap. However, several participants performed a bimanual manipulation with a second grasp to extract the garment and overcome this issue. Another strategy observed was the participants' use of the environment or their bodies to assist with fabric manipulation, i.e. folding the garment over an arm or laying it out on the table. While not directly impacting gripper design, these strategies indicate that adaptation methods may be a viable approach in a deployed system when a grasp configuration or robot strength is limited.

This chapter now presents statistical interpretations of the task duration data. Table A.1 in Appendix A.3.1 contains the collected task timings of the participants completing the set of instructions. Three participants failed to follow instructions correctly during the data collection and missed specific steps. The missing steps are highlighted in Table A.1. For the duration analysis, participants who made errors are omitted. Thus this step uses the duration data from 17 participants. The duration data covers the tasks of folding a shirt, sorting garments, and sorting with folding garments and compares these times under the three constraints. A visualisation of the data as histograms is presented in Figure 3.21, and as a box and whisker plot in Figure 3.22. For folding a T-shirt, these visualisations show the average duration of the three task attempts under each constraint. Both Figures 3.21 and 3.22 show how the applied constraints usually increased the duration of tasks when folding was involved. However, the task of sorting (Figures 3.21(b) and 3.22(b)) saw a decrease in task duration when constraints were applied when compared to the unconstrained condition. Table 3.2 presents the mean, median, and mode of the duration data across these tasks.

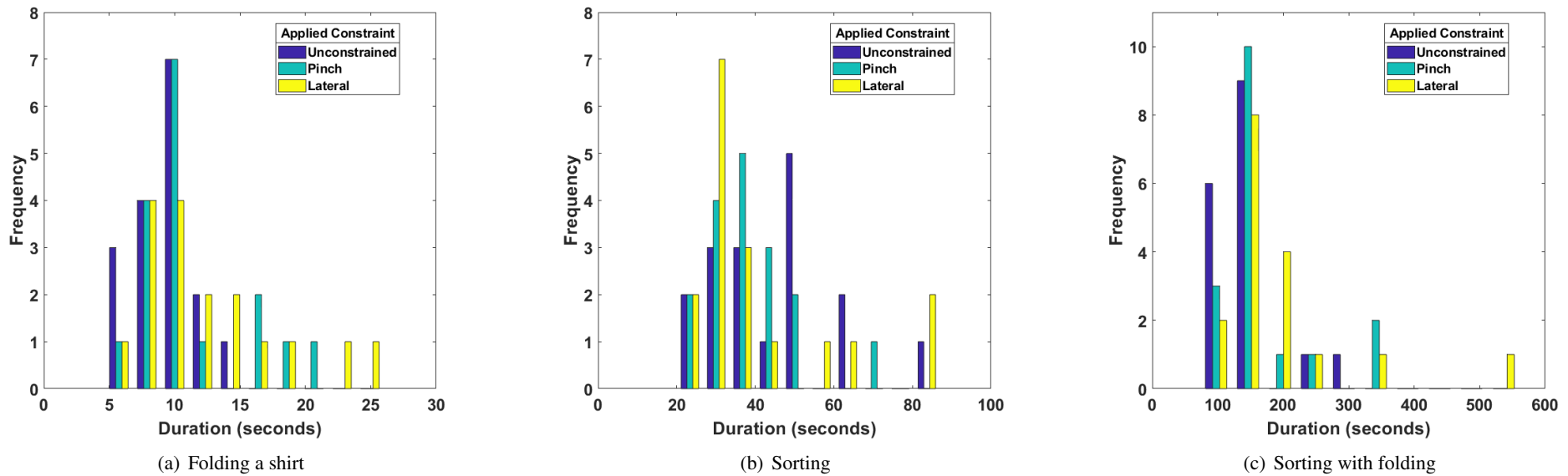


Figure 3.21: Histograms of the time-taken while performing several manipulation tasks under the three constraints.

Additional notes:

These histograms uses measurements of the time-taken by participants and groups the data-points into clusters within a range of 2.2 seconds. After this process, these graphs plot the frequency of data-points that fall into each cluster.

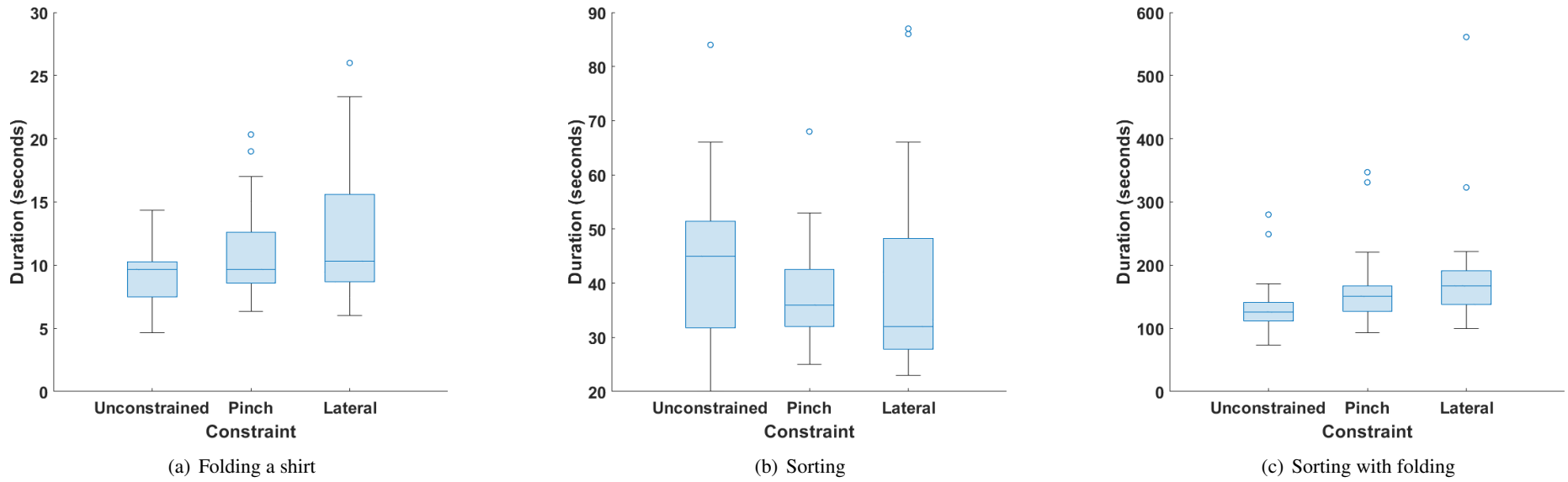


Figure 3.22: Box-plots of the time-taken while performing several manipulation tasks under the three constraints.

Table 3.1: Friedman test with Durbin-Conover pairwise comparisons (generated from the Jamovi software).

Table 3.1.1: Folding T-shirt

Friedman		
χ^2	df	p
14.2	2	< .001

Pairwise Comparisons (Durbin-Conover)

		Statistic	p
Unconstrained	- Pincer	2.57	0.015
Unconstrained	- Lateral	4.80	< .001
Pincer	- Lateral	2.23	0.033

Table 3.1.2: Sorting fabric

Friedman		
χ^2	df	p
5.12	2	0.077

Pairwise Comparisons (Durbin-Conover)

		Statistic	p
Unconstrained	- Pincer	2.0157	0.052
Unconstrained	- Lateral	2.1073	0.043
Pincer	- Lateral	0.0916	0.928

Table 3.1.3: Sorting with folding

Friedman		
χ^2	df	p
12.1	2	0.002

Pairwise Comparisons (Durbin-Conover)

		Statistic	p
Unconstrained	- Pincer	2.70	0.011
Unconstrained	- Lateral	4.15	< .001
Pincer	- Lateral	1.45	0.156

Table 3.2: The mean, median, and mode information for the duration (seconds) of various tasks under different constraints. *UC* refers to unconstrained behaviour.

	Folding a shirt			Sorting			Folding and sorting		
	<i>UC</i>	Pinch	Lateral	<i>UC</i>	Pinch	Lateral	<i>UC</i>	Pinch	Lateral
Mean	9	11	13	45	38	41	139	167	190
Median	10	10	10	45	36	32	126	151	167
Mode	10	9	8	51	36	23	134	93	142

Table 3.1 displays the statistical evaluation results for the three tasks. Values of probability (p) less than or equal to 0.05 indicate a rejection of the null hypothesis that *There is no difference regarding temporal efficiency between constraints while manipulating fabric in various tasks*, thereby indicating a statistically significant difference between some constraint pairs. The results from Table 3.1 showed a statistically significant differences when folding a shirt and sorting with folding. Furthermore, deeper examination of the Durbin-Conover pairwise evaluations shows statistically significant differences in the data are present between all three constraint comparisons under the folding a shirt task. For sorting with folding, statistically significant differences were present for both the unconstrained comparisons, while no difference was present between the pincer and lateral constraints. Such observations reinforce the prediction of folding becoming more complex when only two virtual fingers can be used. Finally, the only pairwise significant difference observed for sorting was between the unconstrained and lateral constraint comparison. Interestingly, the average time of the lateral condition when performing the sorting task was approximately 3 seconds less than the unconstrained condition.

The measurements of central tendency (Table 3.2) indicated that tasks involving folding showed a generally increased average duration when both constraints were applied, with the lateral constraint also further increasing the duration compared to the pinch constraint. One could attribute such a trend to the fact that folding fabric is complex when using only two virtual fingers, i.e., the pinch and lateral constraints. Donaire *et al.* [Donaire 20] specifically mention how folding only using pinch grasps can be difficult. However, the sorting task saw a decrease in average duration when the pinch constraint was applied, and the lateral average duration was higher than the pinch but lower than the unconstrained scenario. One could attribute this decrease in average duration while sorting to the within-subject nature of this experiment and the fact that sorting fabrics is a skill only requiring point-to-point grasps, meaning the task was not hindered by constraints. As participants repeated the various instructions in the same order, they may have gotten used to the tasks and repetition under the constraints. Therefore, participants would not have been hindered by initial hesitations. However, the duration data indicates that humans

Table 3.3: Paired Samples T-test with Wilcoxon rank evaluation - Participants response on grasp constraint difficulty.

Paired Samples T-Test

			Statistic	df	p	Mean difference	SE difference	95% Confidence Interval	
								Lower	Upper
Pincer_folding	Lateral_folding	Student's t	-5.00	19.0	< .001	-34.3	6.85	-48.6	-19.92
		Wilcoxon W	17.00		0.001	-37.5	6.85	-50.0	-20.0
Pincer_unfolding	Lateral_unfolding	Student's t	-3.32	19.0	0.004	-26.4	7.95	-43.0	-9.76
		Wilcoxon W	1.00 ^a		0.002	-30.0	7.95	-65.0	-20.0
Pincer_grasping	Lateral_grasping	Student's t	-2.56	19.0	0.019	-22.3	8.71	-40.5	-4.06
		Wilcoxon W	18.00 ^b		0.003	-30.0	8.71	-40.0	-20.0

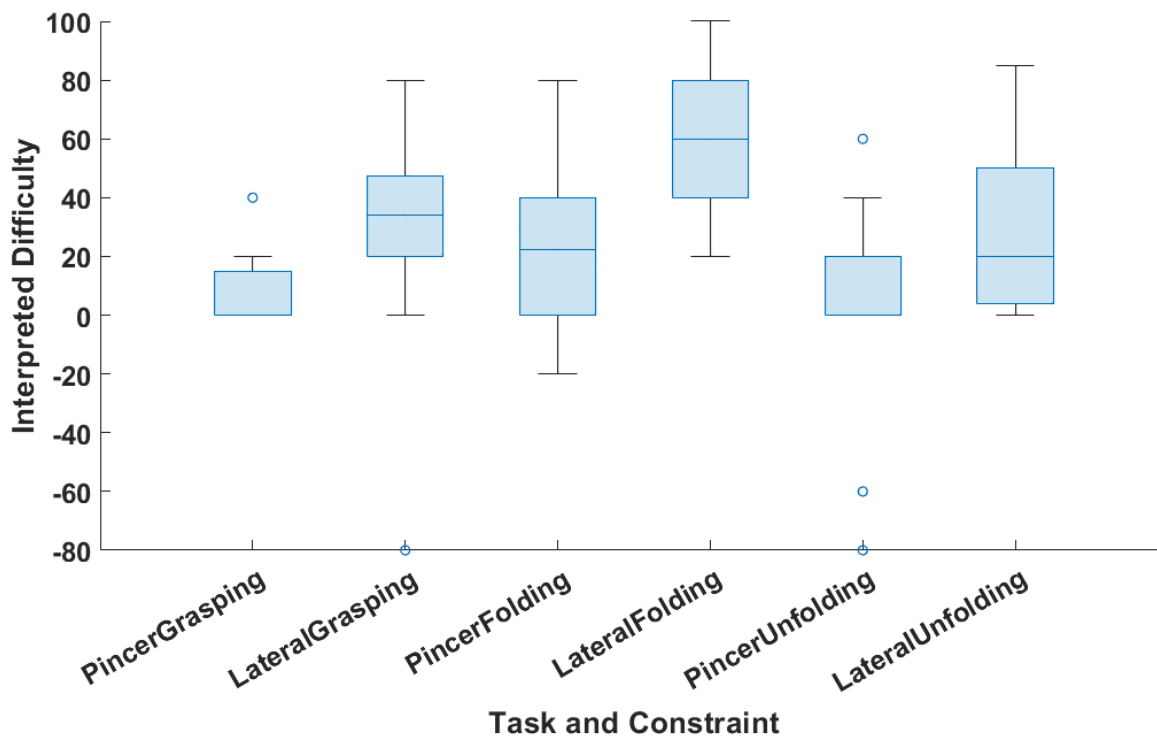
^a 7 pair(s) of values were tied^b 2 pair(s) of values were tied

Figure 3.23: A visualisation of participant responses to how difficult tasks under constraints were when compared to the unconstrained condition.

could perform basic textile sorting at similar temporal efficiencies regardless of constraints.

As previously mentioned, participants filled out a survey upon completing the tasks under various constraints (Appendix A.2; collected data is available in Appendix A.3.1). Figure 3.23 provides a box and whisker plot that visualises the participant response data. These results as analysed in Table 3.3, indicate that participants saw the lateral constraint as more difficult than the pinch constraint across the tasks of grasping, folding and unfolding. The task with the strongest rejection of the null hypothesis (that constraints do not impact manipulation difficulty) was folding, further reinforcing how more than two virtual fingers should be considered for folding-based manipulation primitives.

3.5.4 Discussion

A number of thought-provoking observations present themselves when considering the anthropomorphic hand-centric discussion alongside the user study. Firstly, humans can exploit the environment to grasp fabric regardless of constraints. Additionally, as constraints are applied, humans can still complete all manipulation tasks, including folding, sorting and grasping. However, participants did not respond positively to the lateral constraint. Feedback from the fourth query in the questionnaire (see Section A.2) indicated that participants felt a mental block while under this constraint. Several participants mentioned that this approach felt unfamiliar or counter-intuitive, while others detailed additional motion planning steps to move to an ideal configuration under the lateral constraint.

To reiterate, the primary objective of this project is fabric sorting or pick-and-place applications and the target application of the novel gripper is the ability to grasp and sort textile waste streams. While participants found difficulties using the lateral grasp, it remains a simple mechanical configuration that could be useful for sorting applications. Grippers using the full capabilities of the human hand or the pincer grasp could remain complex if replicated in a gripper design, whereas the lateral grasp mostly uses the motions of the thumb to press into the side of the index finger. Regarding efficiency, with human subjects the lateral constraint took longer on average than the pincer constraint for the tasks of folding, sorting and sorting with folding. However, sorting fabric into various categories saw no statistically significant difference between pincer and lateral constraints, and the average task completion time range while sorting was within six seconds across all task attempts under the various constraints. Furthermore, the literature suggests that no existing gripper design for fabric manipulation is inspired directly by the lateral grasp, thus, such a design offers a novel potential contribution to the field of gripper design.

Another aspect to note is the taxonomy discussion on single-arm grasp force requirements. Previous

devices indicate that around $5N$ for generalised fabric manipulation would suffice for basic tasks, and for extreme extraction scenarios $30N$ of grip strength. However, when participants found their grasp strength or kinematic configuration insufficient in the user study, a second hand was applied to assist with the extraction task. For a single gripper design, these factors indicate that the novel device detailed in Chapter 4 should still target a maximum grasp strength of $30N$. Additionally, sensors that identify material slippage and grasp strength modulation are likely to be beneficial.

3.6 A Refined Scope for a Unique Manipulator Solution

From the taxonomy analysis and user study, the requirements for a unique gripper for fabric manipulation based on the lateral grasp are becoming apparent. They further indicate that the manipulator should be capable of a grasping force of $30N$, but that a reduced grasp force may be sufficient for manipulation actions outside of entangled extraction. Multiple parameters can influence grasp strength and behaviour, including fingertip friction or shape and the material properties of the held garment. Therefore, evaluating the manipulator's grip strength should include holding various fabrics with increasing payloads. Another desired feature is the modulation of the manipulator's grasp strength to enable tasks such as grasp gaiting or haptic exploration. While the haptic exploration feature is not examined in this thesis, enabling the degrees of freedom to explore this feature is still a requirement. The assumption is that for the simple task of fabric sorting, any point-to-point grasp should remain sufficient and not impact the efficiency of the task.

The primary research gap observed in current manipulators is that no biomimetic grasping method can occur from various wrist orientations. This gap partially drives the development of the novel manipulator described in Chapter 4. Once built and attached to a robot arm, the manipulator needs to be able to pin flattened garments to a surface from a range of wrist orientations and execute a biomimetic grasp. Additionally, while performing this grasping motion, the proposed manipulator is expected to be able to control the wrench exerted at the fingertip, which will drag across the environment surface. This requirement derives from the practical difficulties encountered by Koustoumpardis *et al.* [Koustoumpardis 14], in which the friction of the fingertip impacted actuator torques while executing the biomimetic grasping action, sometimes failing to grasp the material from a surface. It can be challenging to overcome these issues while executing a grasp in this manner, as the precise details of the grasped fabric, table surface parameters and the fingertip create a complex physical interaction. Using a force control mechanism may

better account for unpredictable physical interactions, including friction, and currently stands as a novel approach to this problem. Finally, while developing this manipulator, a target aspiration is to remain a simple mechanical system that is a serial-chain manipulator, while retaining a wide range of dexterous skills.

By studying previous devices described the literature with anthropomorphic taxonomies and analysing the user study, this chapter highlights gaps which can inform novel approaches toward gripper design within deformable manipulation. As discussed in the survey into grasping by Babin and Goselin [Babin 21], formulating robotic grasping objectives with taxonomies remains common in the literature. However, this chapter's investigation into fabric manipulation using anthropomorphic classification schemes has presented a different perspective, finding several previously unaddressed gaps. Furthermore, considering previous devices' desired applications and underlying technical details alongside this human-inspired perspective can lead to unexplored design avenues. The formulation of these observations was made possible by following the double diamond design thinking process steps of *define* and *ideate*. Chapter 4 will conclude the process by addressing the *prototype* and *deliver* steps.

Chapter 4

A Novel Manipulator for Fabric Manipulation

This chapter builds upon the human-inspired gripper design and requirements outlined in Section 3.6 by creating a manipulator inspired by the lateral grasp, an anthropomorphic grasp pose visualised in Figure 4.1. Following the multidisciplinary approach outlined in Section 1.4 and the research of Chapter 3, this chapter translates morphological aspects of the lateral grasp into a serial manipulator design that is capable of dexterous textile waste handling. The methodology of this chapter uses classical mechanics, fabrication techniques, and engineering practices to iteratively develop the manipulator, which results in a simulated URDF model alongside a hardware prototype as shown in Figure 4.2. The requirements outlined in Chapter 3 indicate that the maximum grasping force required for edge cases in fabric manip-

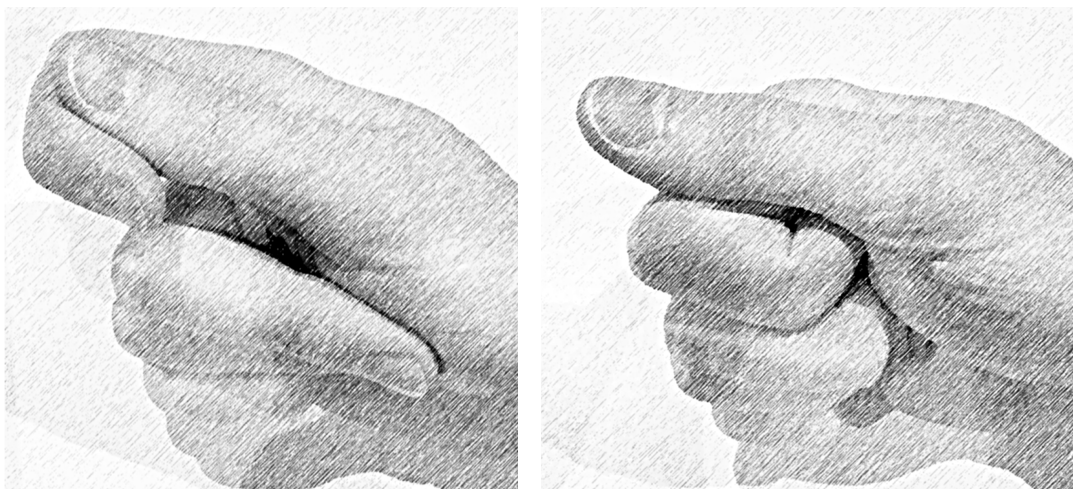


Figure 4.1: The lateral grasp in the poses Lateral Extension Grasp (LEG), left, and Lateral Flexion Grasp (LFG), right.

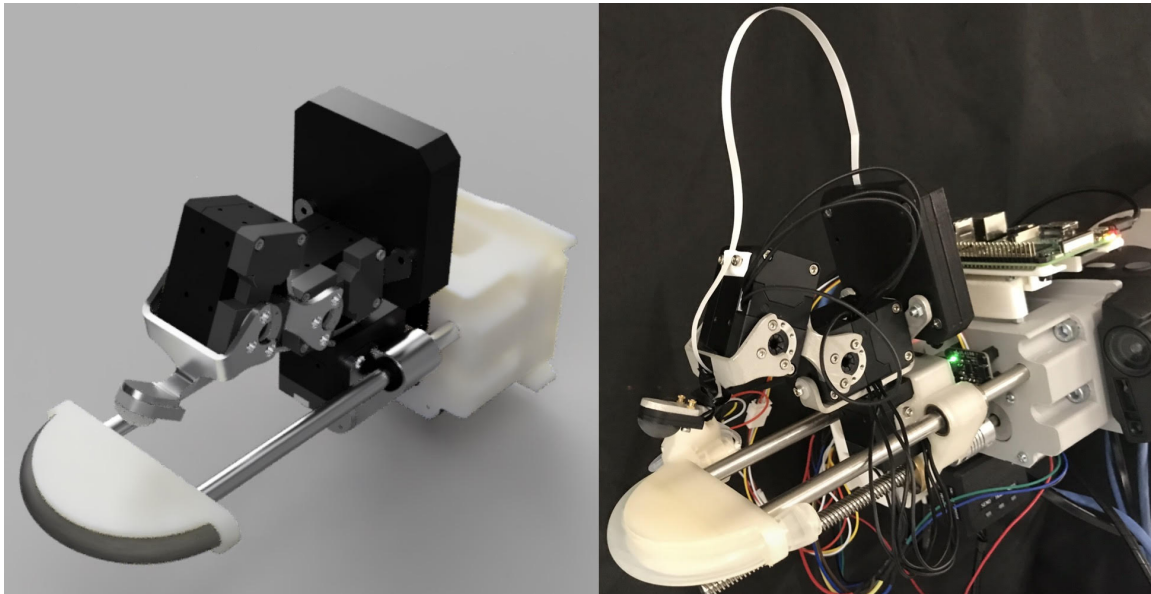


Figure 4.2: A rendering of the gripper prototype alongside the developed hardware platform.

ulation is around $30N$, as discussed by Le *et al.* [Le 13]. However, the gripper also needs to modulate the applied grasp force. Additionally, this device needs to be capable of pinning flattened fabrics to an environmental surface from an arbitrary range of wrist orientations before performing the act of environmentally constrained (*EC*) grasping, which Chapter 5 explores further with data-driven approaches. When considering the *Double Diamond Design Thinking* framework, this chapter performs the steps of *develop (prototype)* and *deliver*, resulting in a fabricated device ready for data-driven experimentation. Figure 4.3 presents the hardware platform with the key components labelled for reference. For further visualisation, Section 4.3 contains an orthogonal engineering drawing of the gripper (Figure 4.10) alongside an exploded assembly diagram (Figure 4.11).

Zwart [Zwart 22] presents an overview of engineering epistemology, the study of how engineers address ‘wicked’ challenges. ‘Wicked’ engineering problems refer to complex and ambiguous challenges that are difficult to define, requiring both theoretical and practical thinking patterns. Translating a grasp defined in neuroscientific literature into a robotic manipulator can be considered a ‘wicked’ challenge, as constructing a robot manipulator is a multidisciplinary undertaking that incorporates aspects of mechanics, electronics and design while still creating a system that is faithful to the original design scope. This chapter is an attempt to address the epistemological question: *Can one use robotic development techniques to build and integrate a unique gripper based structurally around the lateral grasp?* To address this query, a joint effort between University of Canberra collaborators was required. For a complete list of project contributors, please see Appendix C.1.

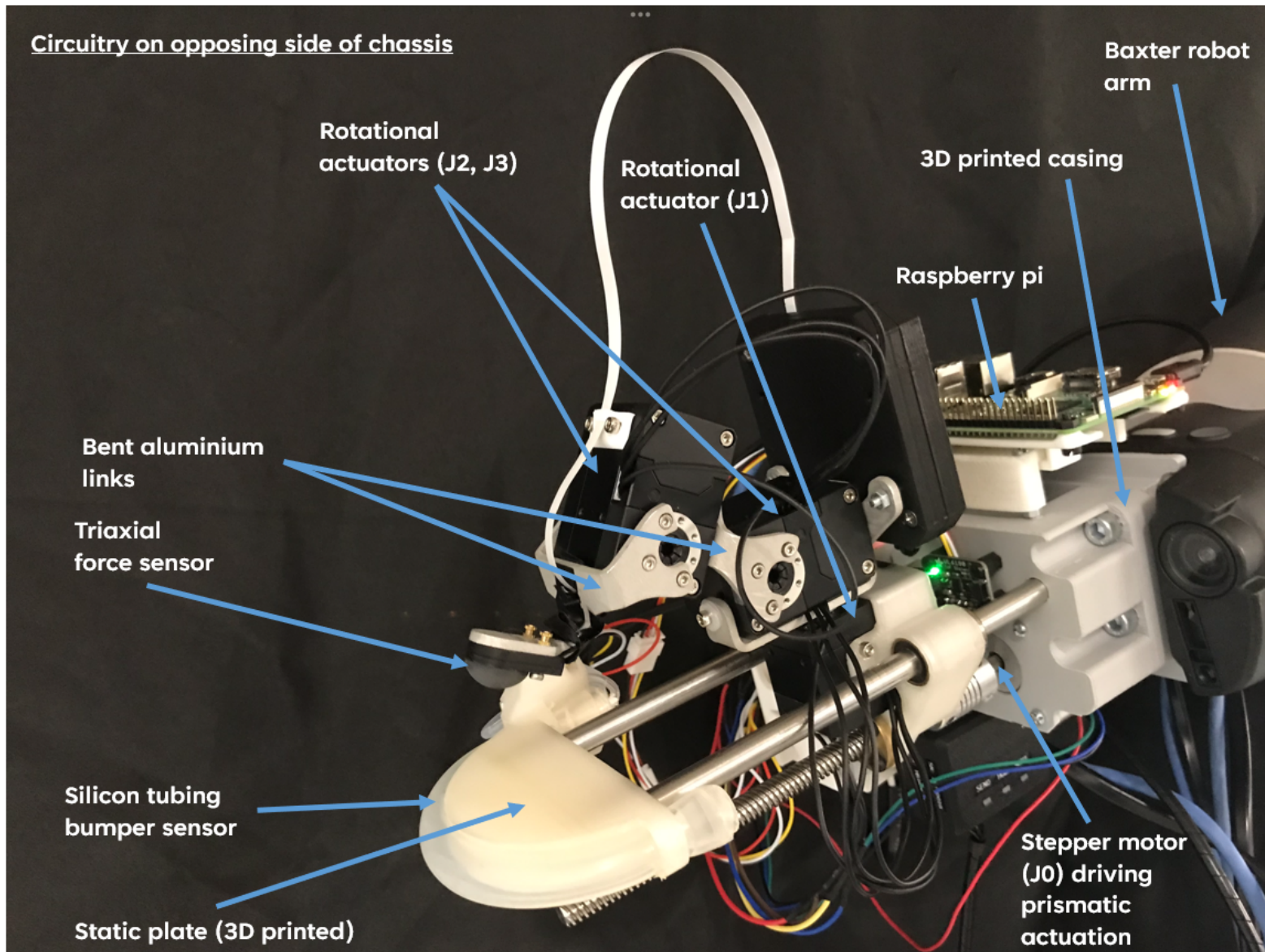
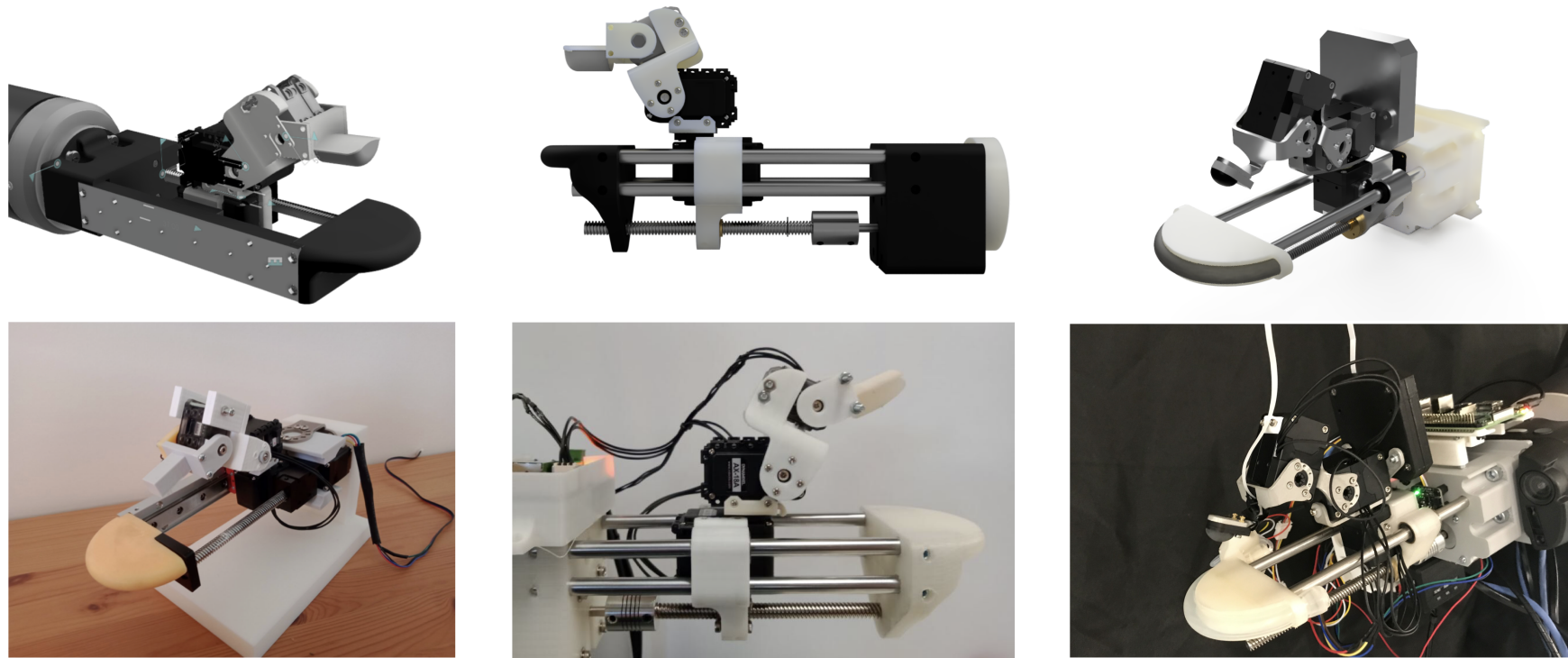


Figure 4.3: The gripper with components labelled.

A range of development information is present to answer this query. Firstly, the content starts by outlining a mathematical model, which includes analyses of the device’s kinematics, kinostatics and dynamics. Such a process moves the device from a conceptual design towards a realised mechanical system. Alongside the model, this chapter will also discuss integration details, including hardware construction, sensor integration, ROS (Robot Operating System) communication, firmware development, control mechanisms and simulation construction for further exploration with deep reinforcement learning (RL) algorithms. A preliminary survey also evaluates the manipulator’s grasp strength and capability to hold various materials.

4.1 Conceptual Design

The lateral grasp, Figure 4.1, is an intermediate type grip that contains elements of both power and precision. The grasp points of contact are between the adducted thumb and the side of the index finger. Under the original granularity of the GRASP taxonomy, such a description significantly distinguishes this configuration from alternative grasps. However, upon closer inspection, the lateral grasp can exist within a range of hand-pose configurations. Figure 4.1 illustrates two sub-configurations of the lateral grasp, denoted as the lateral extension grasp (LEG) and the lateral flexion grasp (LFG). The key differences between these configurations include the rotation of the index finger’s metacarpophalangeal (MCP) joint, which extends or retracts the position of the index finger for contact. When in the LFG configuration, the thumb can exert a greater force against the index finger if considering the pose as a two-link planar manipulator. While in the LEG configuration, the lateral grasp exerts a weaker grasp strength, but more of the thumb’s fingertip can remain in contact with the index finger. This information infers that the proposed manipulator can partially modulate the grasping strength based on the kinematic configuration. The lateral configuration also remains a pose that performs *point-to-point* grasps. However, if considering the geometric definitions of Borràs *et al.* [Borràs 20] while observing Figure 4.2, one could define the grasps of the proposed gripper as *point-to-plane* grasps.



(a) The first conceptual gripper prototype using FDM components on a mount.

(b) A second gripper prototype with an updated rail design and larger stepper motor.

(c) The final iteration using aluminium links with the appropriate impedance actuators and sensors.

Figure 4.4: The iterations of the gripper across development.

The lateral grasp is easy to describe, with the thumb performing most of the required motions. Thus one can represent a robotic equivalent as a serial-chain mechanism. The formulated gripper is a sequential joint-link device with a prismatic joint followed by three rotational joints, which broadly mimic the motions of a human thumb. A static plate on the front of the manipulator represents the side of the index finger, and the prismatic rail can move the thumb appendage closer and further away, replicating the range of motion between the LEG and LFG (Figure 4.1). Therefore, one could consider the initial prismatic rail component as a substitute for the index finger's MCP joint rotation. The static plate is a hemispherical shape with an embedded bumper sensor that detects collisions with the environment. This plate also pins fabric to surfaces, enabling biomimetic grasping motions, and its shape enables the pinning act to occur from a range of wrist orientations. The thumb appendage presses down upon the plate, which acts as the second virtual finger, to hold the fabric in a point-to-point grasp pose.

Following the iterative nature of prototyping within **Design Thinking**, developing the gripper was an iterative process that underwent several redesigns while establishing a viable research prototype, as shown in Figure 4.4. The first prototype, Figure 4.4(a), initially used position-controlled Dynamixel actuators to validate the gripper design. The publication of Hinwood *et al.* [Hinwood 20] presents this first prototype and demonstrates the manipulator moving and performing basic grasps that leverage the environment. Following this design, a second iteration presented an improved rail system that drives the prismatic actuator, shown in Figure 4.4(b). This second iteration validated the final geometry for the gripper prototype while highlighting remaining aspects to update. Finally, Figure 4.4(c) displays the final minimum viable product (MVP), which uses servomotors with impedance controllers. The geometries established in the second iteration informed the choice to implement these actuators. In addition, the MVP also included sensing devices crucial to tasks involved in fabric manipulation.

4.1.1 Gripper Sensors

Many grippers discussed in Chapters 2 and 3 used sensors to detect forces at the fingertips or infer information from haptic exploration. The NAIST M2S gripper [Von Drigalski 17b] used two triaxial force sensors in the fingertips to measure grasp forces and gather feedback for haptic exploration. Alternatively, the sensor developed by Denei *et al.* [Denei 17] for fabric manipulation is a multi-modal device consisting of a capacitive pressure sensor array, microphone, and ambient light sensor. Such a device measures grasp forces, provides feedback from haptic exploration and determines grasp success. When implementing the proposed gripper, two sensors interact with the environment while providing feedback.

The first sensor is a collision detection sensor placed on the front of the static plate. A silicon tube with an attached pressure sensor provides feedback to the gripper system, indicating when the gripper collides with a surface in the environment.

The proposed gripper also contains a triaxial force sensor developed by Contactile¹ embedded in the fingertip. These sensors result from research by Khamis *et al.* [Khamis 18, Khamis 19]. The sensor consists of a rubber body, with a pinhole camera and quadrant photo-diode embedded in the base. Inside the rubber body, a diffuse reflector with illumination LEDs is present. As contact occurs, the rubber body moves and deforms, and the diffuse reflector's positioning shifts while the photo-diode reads light data from the diffuse reflector.

Through a calibration and mapping sequence, this setup allows the sensor to read forces accurately in three directions. Description of the electronics, calibration sequence and accuracy details are provided by Khamis *et al.* [Khamis 19]. The sensor can measure forces of up to $35N$ along the z -axis and $\pm 7.5N$ along the x -axis and y -axis. However, a limitation of this sensing device is that the calibration process occurs at the centre of the sensor's rubber component. As a result, the accuracy declines as the contact interaction moves away from this centre point. In discussions with the manufacturer, this degradation is under investigation, but a solution to accurately model the phenomena in simulation is presently unavailable. Figure 4.5 visualises the dimensions and key technical aspects of the sensor. This sensor can also be seen at the end of the final link of Figure 4.2, and Figure 4.3 explicitly outlines the position of the sensor.

¹<https://contactile.com/>

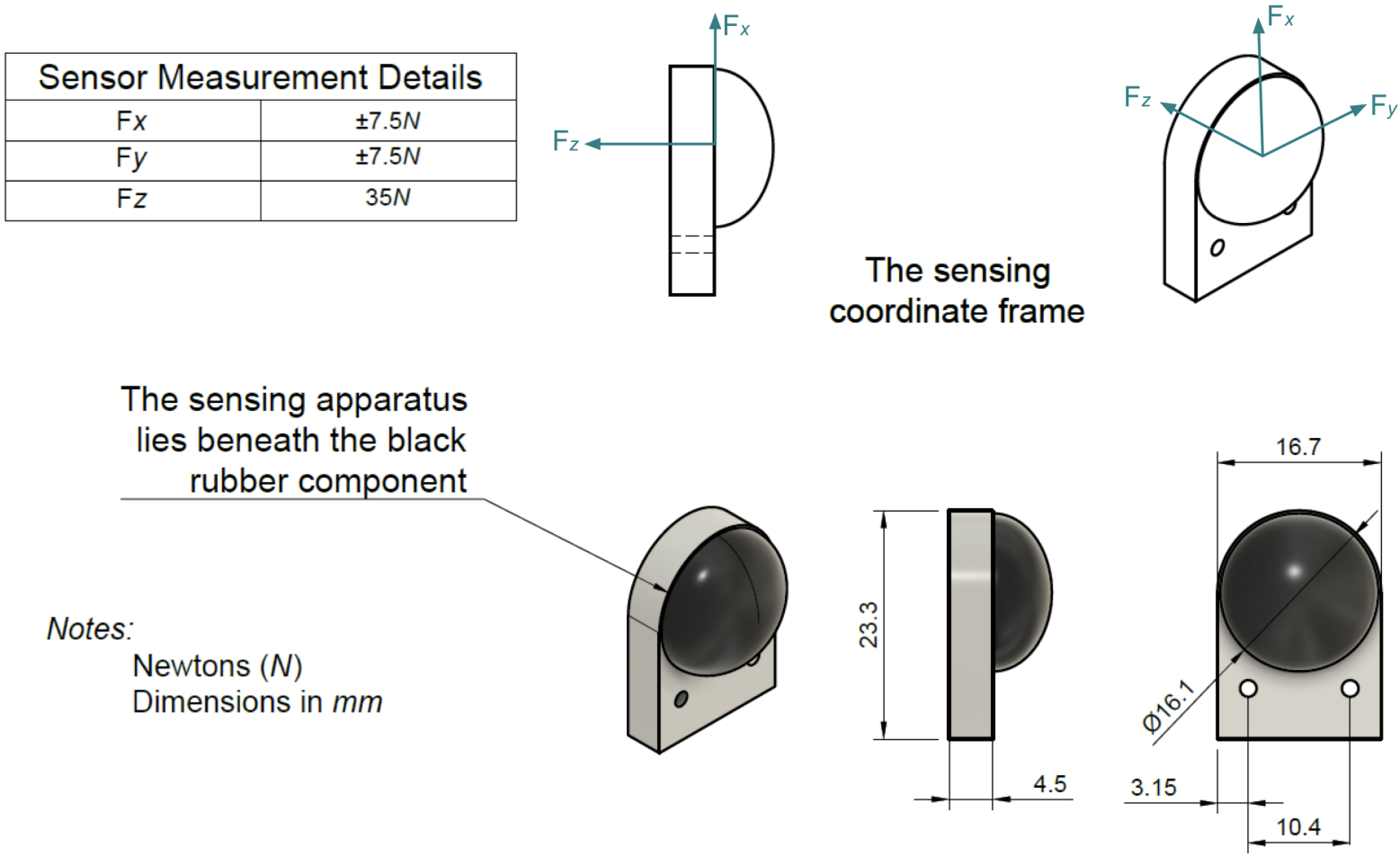


Figure 4.5: Dimensions and sensing coordinate frame of the triaxial force sensor.

4.2 System Model

The gripper described in this chapter is an underactuated serial-chain manipulator with 4 degrees of freedom (DoF) and four actuators. Such a design results in a limited range of possible tool-centre-point (TCP, the end of the serial chain) configurations. This section presents the system model, which details the kinematic, kinostatic and dynamic analyses, informing gripper operation and actuator choice. The kinematic analysis establishes a relationship between the TCP and actuator's motion variables (i.e. position and velocity), while the kinostatic analysis relates the wrench of the TCP to the actuator forces. Finally, the dynamic analysis establishes the required forces while considering the motion parameters of position, velocity and acceleration. In order to minimise complexity, this modelling process does not consider tendon-driven components and custom gearing mechanisms. The gripper model also considers an alternative TCP that reflects the triaxial force sensor's position. Appendix B contains MATLAB code which performs the required calculations for the modelling described throughout this section.

4.2.1 Kinematic Analysis

Kinematic analyses in the context of serial-chain devices refer to a series of numerical or algebraic techniques that relate TCP features to actuator behaviour. The initial step solves a specific set of non-linear equations that relate the TCP's position and orientation to the actuator configuration [Nielsen 99]. In the case of a serial structure, elementary transforms sequentially apply from the base position until reaching the final coordinate frame or TCP. Figure 4.6 visualises the transforms of the proposed gripper with frames w and e representing the base and TCP, respectively. Given the limited actuators in the gripper, both the forward and inverse kinematic equations are solvable via a closed-form algebraic solution.

Position Kinematics

The initial step in modelling the gripper as a serial-link manipulator begins with the standard Denavit-Hartenberg (DH) parameters defined by Corke [Corke 17]. The DH parameters are a concise table format that presents the sequential transforms of a serial-link structure. Table 4.1 details the parameters of the manipulator. Included values for length parameters in Table 4.1 represent information from the final prototype designed in CAD software. Additionally, a base transform is required to orient the manipulator along the x -axis of the local world frame w . For brevity, this base transform is present in the first row of Table 4.1. Figure 4.6 visualises the DH parameters with the blue coordinate frame representing the local

world frame (w) and the red coordinate frame representing the TCP (e). For future reference, the following kinematics map actuator positions and speeds to the same features of frame e . Before proceeding with the modelling process, this chapter also provides some notation details. The transformation A_i indicates the completed transformations of the i th row of Table 4.1, see Equation 4.1. Within this convention, the constant base transform A_w refers to the first row of the DH parameters, using the static operations $R_z(\theta_w)$ and $R_x(\alpha_w)$. Alongside the transformation parameters, another column ($qlim$) denotes the joint range for each actuator (q_i). Figure 4.7 shows a mechanical drawing of the final prototype highlighting the range of motion informed by the $qlim$ values.

Table 4.1: Denavit-Hartenberg parameters

j_i	θ_i	d_i	a_i	α_i	$qlim_i$
\dot{j}_w	$\frac{\pi}{2}$	0	0	$\frac{\pi}{2}$	N/A
j_0	π	q_0	0	$\frac{\pi}{2}$	0 - 79(mm)
j_1	q_1	$d_1(18.6mm)$	$a_1(20mm)$	$\frac{\pi}{2}$	0.8727 - 2.2689(rad)
\dot{j}_2	q_2	0	$a_2(35.7mm)$	0	0 - $\frac{\pi}{2}$ (rad)
\dot{j}_3	q_3	0	$a_3(44.5mm)$	$-\frac{\pi}{2}$	-1.7453 - 0.0175(rad)

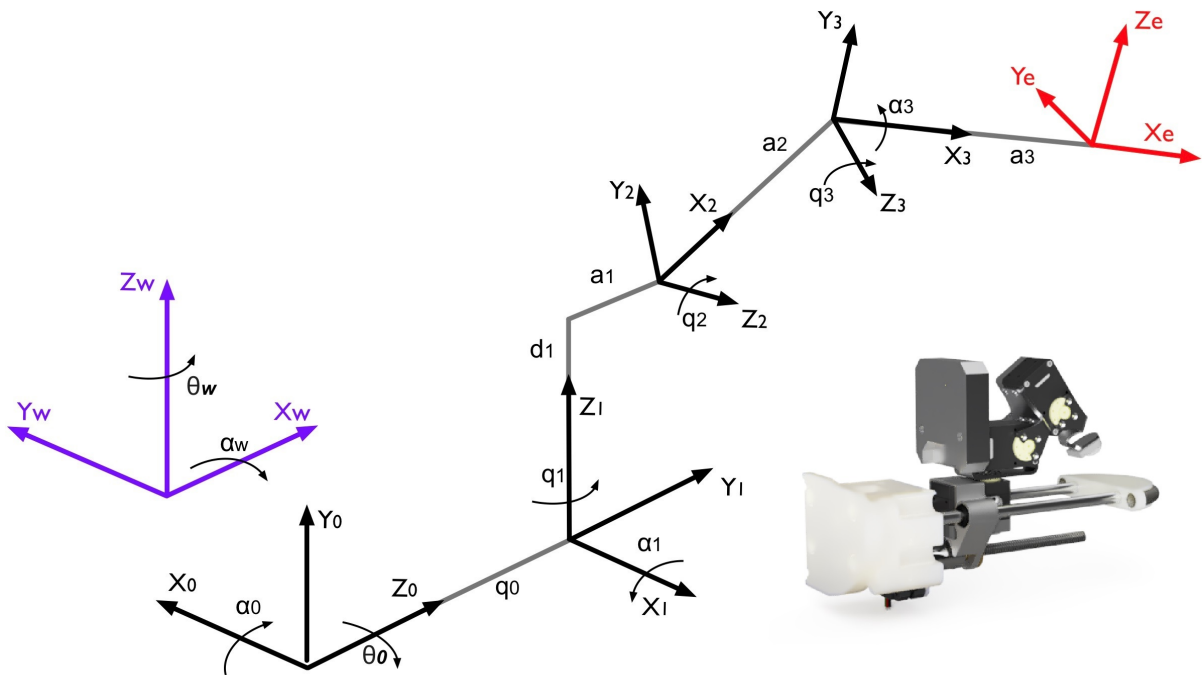


Figure 4.6: Visualisation of the DH parameters alongside a rendered gripper at a similar pose.

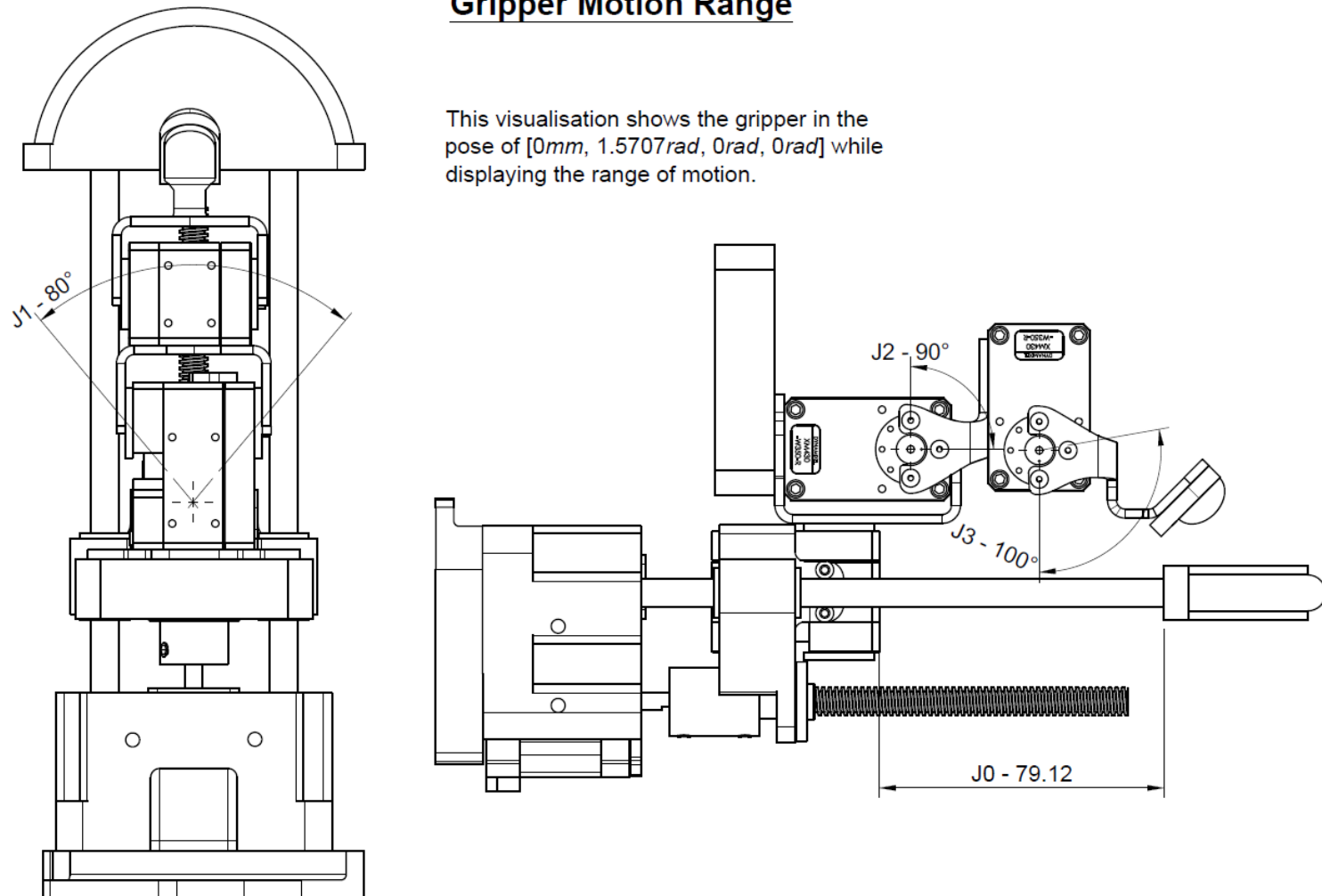


Figure 4.7: The range of motion of the built prototype.

$$A_i = A_{i-1}R_z(\theta_i)T_z(d_i)T_x(a_i)R_x(\alpha_i) \quad (4.1)$$

The forward kinematics calculate the position and orientation of e relative to w using the joint positions. As the proposed manipulator only contains four actuators, the forward kinematics are constrained to represent four TCP elements consisting of the position variables x , y and z alongside the orientation component p (rotation about the y axis) when given the joint configuration q . Equations 4.2, 4.3, 4.4, and 4.5 define these forward kinematic (FK) expressions given the joint positions and define a relationship between the coordinate frames w and e from Figure 4.6.

$$x = q_0 + \sin(q_1) (a_1 + a_3 \cos(q_2 + q_3) + a_2 \cos(q_2)) \quad (4.2)$$

$$y = -\cos(q_1) (a_1 + a_3 \cos(q_2 + q_3) + a_2 \cos(q_2)) \quad (4.3)$$

$$z = d_1 + a_3 \sin(q_2 + q_3) + a_2 \sin(q_2) \quad (4.4)$$

$$p = -q_2 - q_3 \quad (4.5)$$

Assuming that the desired set of positions x , y , z and p are known, one can calculate the inverse kinematics using the closed-form solution of Equations 4.6, 4.7, 4.8, and 4.9. For use during robot operation, one applies these equations in the order of 4.8, 4.9, 4.7 and 4.6.

$$q_0 = x - \sin(q_1) (a_1 + a_3 \cos(q_2 + q_3) + a_2 \cos(q_2)) \quad (4.6)$$

$$q_1 = \pi - \arccos\left(\frac{y}{a_1 + a_3 \cos(q_2 + q_3) + a_2 \cos(q_2)}\right) \quad (4.7)$$

$$q_2 = \arcsin\left(\frac{z - d_1 + a_3 \sin(p)}{a_2}\right) \quad (4.8)$$

$$q_3 = -p - q_2 \quad (4.9)$$

Integrating the Triaxial Force Sensor

Assuming the standard DH convention, the previously presented Equations 4.2-4.9 define how the position and orientation of the TCP interact with the actuators. However, the development of this system requires tactile feedback for reinforcement learning and long-term applications. A triaxial force sensor attached to the TCP provides this functionality, as discussed in Section 4.1.1. As this sensor's position differs from the TCP of the standard DH convention, considerations are made during the kinematic modelling process, resulting in a modified set of forward kinematic equations. A new value defined as Δ_z represents an additional transformation of the z-axis from the DH parameters. When estimated in CAD software, this value was $-13.36mm$. Additionally, the pitch constraint previously declared within the forward kinematics now has a constant value of $\frac{\pi}{4}$ added to the rotation of the TCP. Figure 4.8 highlights both of these changes. Finally, Equation 4.10 represents the coordinate frame of the sensor's position and orientation (A_s) given e . A_s also reflects the directions of measurement the sensor uses when taking force readings.

$$A_s = e \cdot T_z(\Delta_z) R_y\left(\frac{\pi}{4}\right) \quad (4.10)$$

Developing the kinematic relationship between the actuator configuration and A_s requires modification of the original FK expressions (Equations 4.2-4.5) to account for the sensor's position and orientation

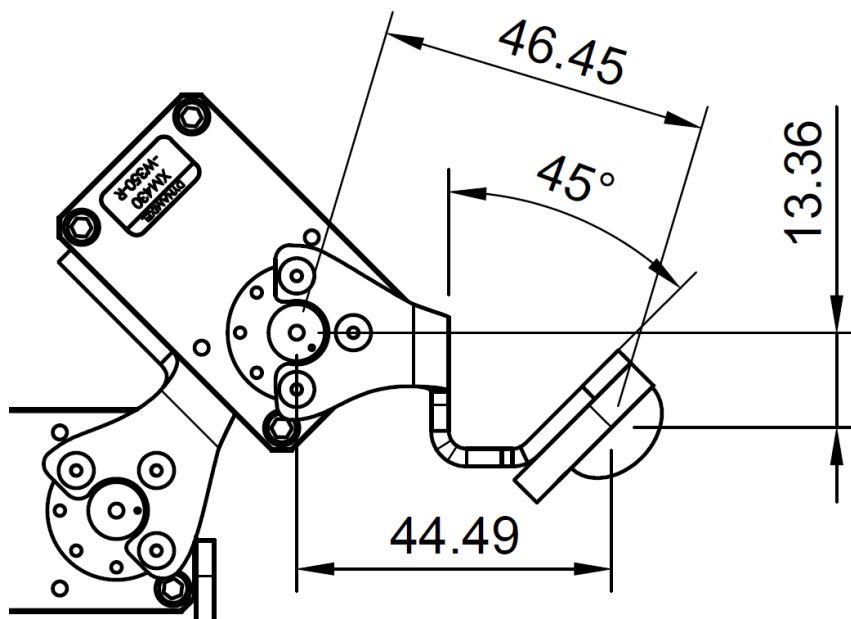


Figure 4.8: Displacement of the triaxial force sensor when attached to the robotic manipulator.

rather than the traditional TCP of the DH parameters. The sensor position deviates from the standard DH convention by the two transformations expressed in Equation 4.10 and visualised in Figure 4.8. The kinematic expressions considering the sensor's position require modified variables, which Equation 4.11 outlines. Figure 4.8 also shows how sensor's orientation requires an offset of $\frac{\pi}{4}$ (45 deg) applied to the calculation of p when considering the sensor. Equations 4.12-4.15 present the modified FK expressions that reflect the features of x , y , z , and p for A_s . These modified expressions are reflected in Appendix B.0.3 which presents the MATLAB code modelling the kinematics of the standard DH parameters alongside the sensor-modified approach.

$$\begin{aligned} \Delta_z &= -13.36\text{mm}, a_3 = 44.49\text{mm} && \text{See Figure 4.8} \\ a_{3\text{alt}} &= 46.45 = \sqrt{\Delta_z^2 + a_3^2} && \text{An alternative } a_3 \text{ value for the sensor (Figure 4.8)} \\ q_O &= 0.2917\text{rad} = \tan^{-1}\left(\frac{\Delta_z}{a_3}\right) && \text{An offset added to } q_3 \text{ for calculation of the sensor's position} \\ q'_3 &= q_3 + q_O && \text{A modified } q_3 \text{ variable which includes the offset } q_O \end{aligned} \quad (4.11)$$

$$x_s = q_0 + \sin(q_1) (a_1 + a_{3\text{alt}} \cos(q_2 + q'_3) + a_2 \cos(q_2)) \quad (4.12)$$

$$y_s = -\cos(q_1) (a_1 + a_{3\text{alt}} \cos(q_2 + q'_3) + a_2 \cos(q_2)) \quad (4.13)$$

$$z_s = d_1 + a_{3\text{alt}} \sin(q_2 + q'_3) + a_2 \sin(q_2) \quad (4.14)$$

$$p_s = -q_2 - q_3 + \frac{\pi}{4} \quad (4.15)$$

Similarly, the inverse kinematics from the sensor's position and orientation rely on the core equations derived from the DH parameters, i.e. Equations 4.6-4.9, with slight adjustments taken from the previous paragraph resulting in Equations 4.16, 4.17, 4.18, and 4.19.

$$q_{0s} = x_s - \sin(q_{1s}) (a_1 + a_{3\text{alt}} \cos(q_{2s} + q_{3s} + q_O) + a_2 \cos(q_{2s})) \quad (4.16)$$

$$q_{1s} = \pi - \arccos\left(\frac{y_s}{a_1 + a_{3\text{alt}} \cos(q_{2s} + q_{3s} + q_O) + a_2 \cos(q_{2s})}\right) \quad (4.17)$$

$$q_{2s} = \text{asin} \left(\frac{z_s - d_1 + a_{3alt} \sin(p_s - \frac{\pi}{4} - q_0)}{a_2} \right) \quad (4.18)$$

$$q_{3s} = -p_s - \frac{\pi}{4} - q_{2s} \quad (4.19)$$

To this point, the presented kinematic analysis uses the traditional DH parameters to define a novel robot manipulator alongside the closed-form algebraic expressions for the forward and inverse kinematics, defining the positional relationship between the TCP's pose and the actuator's configuration. Additionally, the modelling process has modified these expressions to account for the embedded triaxial force sensor's position. Appendix B.0.3 presents these calculations and validates them across the robot manipulator's workspace, assuming the joint position limitations (*qlim*) from Table 4.1.

Velocity Kinematics

The following modelling component builds the mathematical relationship representing velocity between the actuators and TCP based on the positional information obtained in the previous section. The Jacobian matrix discussed by Corke [Corke 17] models this relationship. The Jacobian matrix takes the forward kinematic equations and partially derives each expression by the position of each actuator in the system, resulting in a matrix that acts as a function of actuator positions (q). The final Jacobian, $J(q)$, takes the simplified form shown below in Equation 4.20. Equation 4.21² shows the calculated value of $J(q)$.

$$J(q) = \begin{bmatrix} \frac{\partial x}{\partial q_0} & \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \frac{\partial x}{\partial q_3} \\ \frac{\partial y}{\partial q_0} & \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \frac{\partial y}{\partial q_3} \\ \frac{\partial z}{\partial q_0} & \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \frac{\partial z}{\partial q_3} \\ \frac{\partial p}{\partial q_0} & \frac{\partial p}{\partial q_1} & \frac{\partial p}{\partial q_2} & \frac{\partial p}{\partial q_3} \end{bmatrix} \quad (4.20)$$

$$\begin{pmatrix} 1 & c(q_1) (a_1 + a_3 c(q_{2+3}) + a_2 c(q_2)) & -s(q_1) (a_3 s(q_{2+3}) + a_2 s(q_2)) & -a_3 s(q_{2+3}) s(q_1) \\ 0 & s(q_1) (a_1 + a_3 c(q_{2+3}) + a_2 c(q_2)) & c(q_1) (a_3 s(q_{2+3}) + a_2 s(q_2)) & a_3 s(q_{2+3}) c(q_1) \\ 0 & 0 & a_3 c(q_{2+3}) + a_2 c(q_2) & a_3 c(q_{2+3}) \\ 0 & 0 & -1 & -1 \end{pmatrix} \quad (4.21)$$

A matrix multiplication operation using the \dot{q} vector, which holds the actuators' velocities, allows one to use $J(q)$ during robot operation. Using this operation, one can estimate the velocities in the directions

²Re: Equation 4.21, $q_{2+3} = q_2 + q_3$, $c = \cos$, and $s = \sin$

defined by the forward kinematic expressions. Because the forward kinematics are in reference to w , the velocity vector $\begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \dot{p} \end{bmatrix}^T$ also relates Cartesian velocities with respect to w . Therefore, one can derive the speeds of the TCP with Equation 4.22. Inverting $J(q)$ allows the modelling process to derive a subsequent expression that estimates the required joint speeds from the desired TCP velocity as shown in Equation 4.23.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{p} \end{bmatrix} = J(q)\dot{q} = J(q) \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \quad (4.22)$$

$$\dot{q} = J(q)^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{p} \end{bmatrix} \quad (4.23)$$

Before moving on to kinostatics, the last component to address in Jacobian development is the determinant, $\det(J(q))$. The calculated value of $\det(J(q))$ is presented in Equation 4.24. As Corke [Corke 17] discusses, when $\det(J(q)) = 0$, the Jacobian loses rank and the expressions described in Equations 4.22 and 4.23 become invalid, as a kinematic singularity has occurred. When considering the *qlim* parameters from Table 4.1, only a single condition creates a singularity, namely when the position of q_2 is equivalent to $\frac{\pi}{2}$. While in operation, applying a small amount of noise to q_2 alleviates such a condition. Figure 4.9 visualises the configuration when this scenario occurs.

$$\det(J(q)) = -a_2 \cos(q_2) \sin(q_1) (a_1 + a_3 \cos(q_2 + q_3) + a_2 \cos(q_2)) \quad (4.24)$$

Kinostatic Analysis

To further model the gripper, understanding how the forces of the actuators impact the wrench exerted at the TCP (or fingertip) is essential. Corke [Corke 17] specifies how the wrench at the TCP of a serial link structure is a vector that details the forces and moments exerted in world frame w . One can map the actuator forces to wrenches exerted at the TCP by modifying $J(q)$, enabling calculation of grasp forces

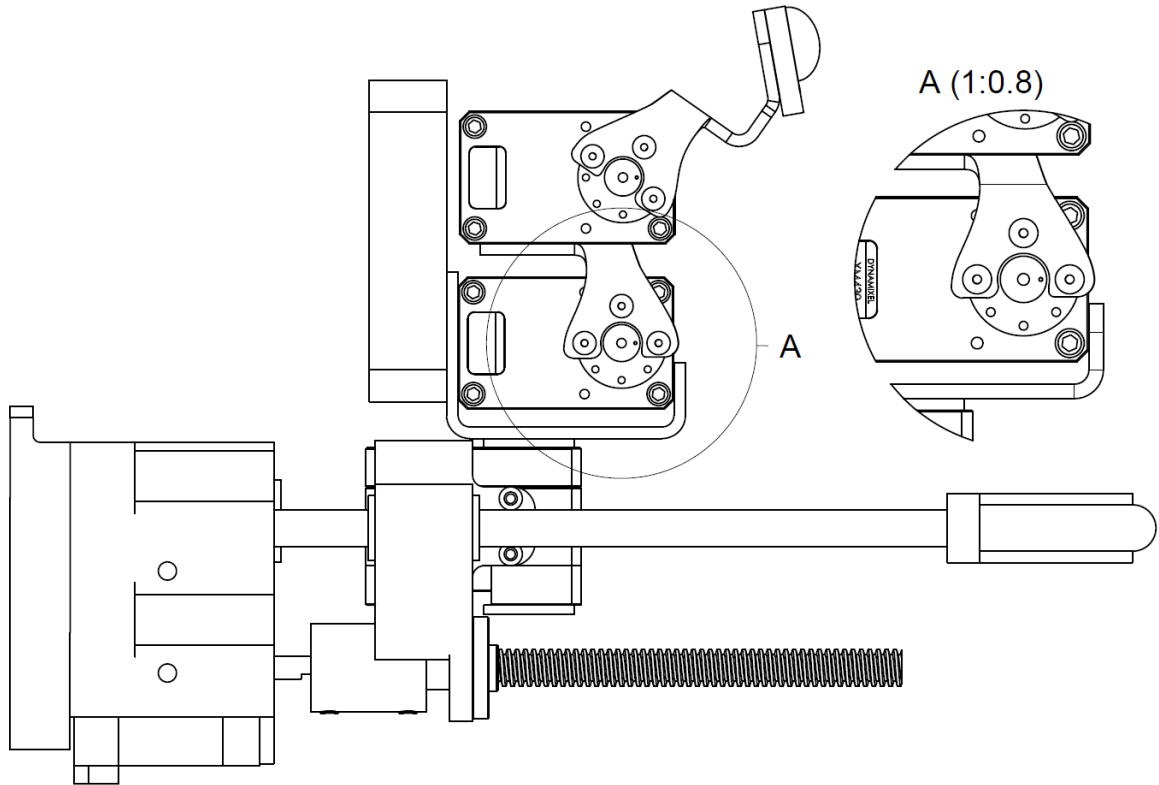


Figure 4.9: A mechanical drawing of the gripper where *detailed view A* shows the position of q_2 (at $\frac{\pi}{2}$) causing a kinematic singularity.

or wrenches applied to the environment. Modelling this process takes $J(q)$ and applies a transpose operation, $J(q)^T$. The wrench at the TCP can then estimate the required actuator forces (τ) (Equation 4.25). One can also perform a similar step shown in the velocity kinematics where $J(q)^T$ is inverted, thus calculating the TCP wrench from actuator forces, shown in Equation 4.26. As the determinant of a transposed matrix is equivalent to the determinant of the original, Equation 4.24 applies to $J(q)^T$.

$$\begin{bmatrix} \tau_0 \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = J(q)^T \begin{bmatrix} f_x \\ f_y \\ f_z \\ m_y \end{bmatrix} \quad (4.25)$$

$$\begin{bmatrix} f_x \\ f_y \\ f_z \\ m_y \end{bmatrix} = (J(q)^T)^{-1} \begin{bmatrix} \tau_0 \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad (4.26)$$

Velocities and Kinostatics of the Sensor

Equation 4.20 develops the Jacobian matrix for the TCP of the standard DH convention. For brevity, the steps to create the Jacobian matrix using the sensor's position ($J_s(q)$) are omitted from this section. However, $J_s(q)$ is calculable simply by using the sensor's forward kinematic parameters (see Equations 4.12-4.15) and using them to construct $J_s(q)$ as shown in Equation 4.27. With this change, the process outlined throughout Equations 4.21-4.26 can be replicated to model the velocity and force relationships assuming a TCP at the sensor's position. The MATLAB code in Appendix B.0.3 contains the calculation of both $J(q)$ and $J_s(q)$.

$$J_s(q) = \begin{bmatrix} \frac{\partial x_s}{\partial q_0} & \frac{\partial x_s}{\partial q_1} & \frac{\partial x_s}{\partial q_2} & \frac{\partial x_s}{\partial q_3} \\ \frac{\partial y_s}{\partial q_0} & \frac{\partial y_s}{\partial q_1} & \frac{\partial y_s}{\partial q_2} & \frac{\partial y_s}{\partial q_3} \\ \frac{\partial z_s}{\partial q_0} & \frac{\partial z_s}{\partial q_1} & \frac{\partial z_s}{\partial q_2} & \frac{\partial z_s}{\partial q_3} \\ \frac{\partial p_s}{\partial q_0} & \frac{\partial p_s}{\partial q_1} & \frac{\partial p_s}{\partial q_2} & \frac{\partial p_s}{\partial q_3} \end{bmatrix} \quad (4.27)$$

4.2.2 Dynamic Analysis

The kinematic analysis of Section 4.2.1 establishes the relationships between the position, velocity and forces applied between w and the TCP as defined by the DH parameters in Table 4.1, alongside a duplicate process considering the integrated triaxial force sensor (Appendix B.0.3). The final modelling component of the gripper is the system's dynamics, establishing the required forces for desired movement. The Lagrangian method formulates the dynamics of the gripper. Such a method establishes the inverse dynamics problem, the estimation of torques and forces considering the position, velocity and acceleration, $q, \dot{q}, \ddot{q} \rightarrow \tau$. Equation 4.28 concisely expresses the dynamics. This representation uses three major components, the joint-space inertia matrix $M(q)$, the Coriolis matrix $C(q, \dot{q})$ and the gravity term $G(q)$, all of which are functions of the gripper's configuration and motion.

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \quad (4.28)$$

As described by Corke [Corke 17], $M(q)$ expresses the inertia experienced by each joint alongside the products of inertia between joints. $C(q, \dot{q})$ is a function of the manipulator's present velocity alongside position, producing information about the centripetal torques caused by each joint and Coriolis torques produced by joint pairs. Finally, $G(q)$ contains information about the forces required to overcome gravity in the manipulator's present configuration. Corke also discusses an additional term of

$F(\dot{q})$, which addresses motor friction. However, details of the friction properties of the actuators utilised throughout the gripper's iterative development were not available; instead they each have an in-built low-level controller through which it's possible to edit specific parameters but not access the underlying mechanisms. Considering these factors, the friction term is absent when modelling the dynamics. Next, symbolic equations manually derive the Lagrangian and validate the calculations with the robotics toolbox of Corke [Corke 17]. Modelling the rigid body equations of motion follows the approaches by Yoshikawa [Yoshikawa 90] or Sciavicco and Siciliano [Sciavicco 12]. The Lagrangian represents the subtraction of potential energy (\mathcal{U}) from kinetic energy (\mathcal{T}) or $\mathcal{L} = \mathcal{T} - \mathcal{U}$. Per a definition from Yoshikawa *et al.* [Yoshikawa 90], Equation 4.29 represents the torque or force required at each actuator (τ_i). Alternatively, the form expressed in Equation 4.30 is also acceptable.

$$\tau_i = \frac{d}{dt} \left(\frac{\partial}{\partial \dot{q}_i} \mathcal{T} \right) - \frac{\partial}{\partial q_i} \mathcal{T} + \frac{\partial}{\partial q_i} \mathcal{U} \quad (4.29)$$

$$\tau_i = \frac{d}{dt} \left(\frac{\partial}{\partial \dot{q}_i} \mathcal{L} \right) - \frac{\partial}{\partial q_i} \mathcal{U} \quad (4.30)$$

The first step is to formulate expressions for T and U to implement this equation. However, to define these terms, dynamic variables are required. These dynamic variables only require the DH parameters of Table 4.1 and treat the sensor as part of the manipulator's final link. For each joint-link pair j_i , there is an associated series of variables, including the inertia tensor I_i , the mass m_i and the centre-of-mass (CoM) P_i . The CoM of each link refers to a position representing the mass of each link within a single particle. In the notation of Corke [Corke 17], this position is from the final transformation of a joint-link pair. Thus if one assumes a constant local CoM transform p_i , the CoM position for j_i relative to w is shown in Equation 4.31 as P_i . Finally, The mass of j_i refers to the weight of the link-pair.

$$P_i = A_i p_i \quad (4.31)$$

I_i for each joint refers to the inertia tensor about each link's CoM (P_i). An inertia tensor takes the form of a square matrix with nine elements as shown in Equation 4.32, per a definition provided by Sciavicco and Siciliano [Sciavicco 12]. The inertia tensor holds the moments of inertia along the diagonal elements I_{xx} , I_{yy} and I_{zz} . The remaining elements present the products of inertia between the axis. Although the values of this matrix for fabricated components can be complex to estimate experimentally, a common approach is to extrapolate the inertia tensor among other dynamic parameters

from CAD software, as mentioned by Gautier and Venture [Gautier 13].

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \quad (4.32)$$

The expressions for kinetic energy (T) and potential energy (U) require these dynamic parameters for calculation. First, the kinetic energy T involves summing the translation and rotation components, $T = T_l + T_r$. Calculating the translation component, T_l , is expressed in Equation 4.33, using the mass m_i and the velocity of the CoM for each link \dot{P}_i .

$$\mathcal{T}_l = \sum_{i=0}^3 \frac{1}{2} m_i \dot{P}_i^T \dot{P}_i \quad (4.33)$$

Calculating \dot{P}_i uses an altered form of the Jacobian matrix shown in Equation 4.34. This method takes the position vector P_i and generates a Jacobian matrix for the CoM of each link i , noting that as shown in Equation 4.34, this Jacobian is always a three-by-four matrix. However, depending on the link calculated, any columns representing actuators beyond the link being evaluated are replaced with zeros. For example, estimating the speed or the CoM of link 0, P_0 would involve only the partial derivative expressions related to q_0 . One would then overwrite the remaining three columns with zeros. For P_1 , the partial derivative expressions related to q_0 and q_1 would occur with the remaining columns then equalling zero. This trend would continue as each joint-link pair is evaluated in the calculation.

$$\dot{P}_i = \begin{bmatrix} \frac{\partial P_{ix}}{\partial q_0} & \dots & \frac{\partial P_{ix}}{\partial q_i} & 0 & \dots \\ \frac{\partial P_{iy}}{\partial q_0} & \dots & \frac{\partial P_{iy}}{\partial q_i} & 0 & \dots \\ \frac{\partial P_{iz}}{\partial q_0} & \dots & \frac{\partial P_{iz}}{\partial q_i} & 0 & \dots \end{bmatrix} \dot{q} \quad (4.34)$$

The second aspect of the kinetic energy calculation, T_r , accounts for the rotational elements within the kinetic energy term. This term uses the constant inertia tensors of each link, I_i , and the rotational velocity of each link, ω_i . While I_i remains constant, ω_i requires calculation. Equation 4.35 expresses the calculation of T_r .

$$\mathcal{T}_r = \sum_{i=0}^3 \frac{1}{2} \omega_i^T I_i \omega_i \quad (4.35)$$

Calculating ω_i involves constructing a rotational Jacobian matrix, which uses derivative values of

rotation matrices (Appendix B.0.2). Such a calculation is also known as the rotational component of the geometric Jacobian. This system's rotational Jacobian $J_R(q)$ results in a three-by-four sized matrix shown in Equation 4.36. A rotation matrix defines each column from the DH convention, $A^{R_{i-1}}$, which defines the rotation matrix transformation up to the last operation before the transformation of joint q_i occurs. The derivative of each matrix is calculable by multiplying the rotation matrix with the constant single-column matrix $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. When a joint is prismatic, the column values default to 0.

$$J_R(q) = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & A^{R_0} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & A^{R_1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & A^{R_2} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} \quad (4.36)$$

To establish ω_i , a similar process shown in Equation 4.34 occurs, in which the irrelevant components involved in calculating the links' velocities take a value of zero. Therefore, Equation 4.37 displays the calculation method for the angular velocity (ω_i) of each link.

$$\omega_i = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \dots & A^{R_{i-1}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & \dots & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} \dot{q} \quad (4.37)$$

The presented calculations establish the kinetic energy $\mathcal{T} = \mathcal{T}_l + \mathcal{T}_r$. However, the potential energy \mathcal{U} still requires calculation. This expression uses the mass and CoM position dynamic parameters with the gravity vector g , as shown in Equation 4.38.

$$\mathcal{U} = \sum_{i=0}^3 (-m_i g^T P_i) \quad (4.38)$$

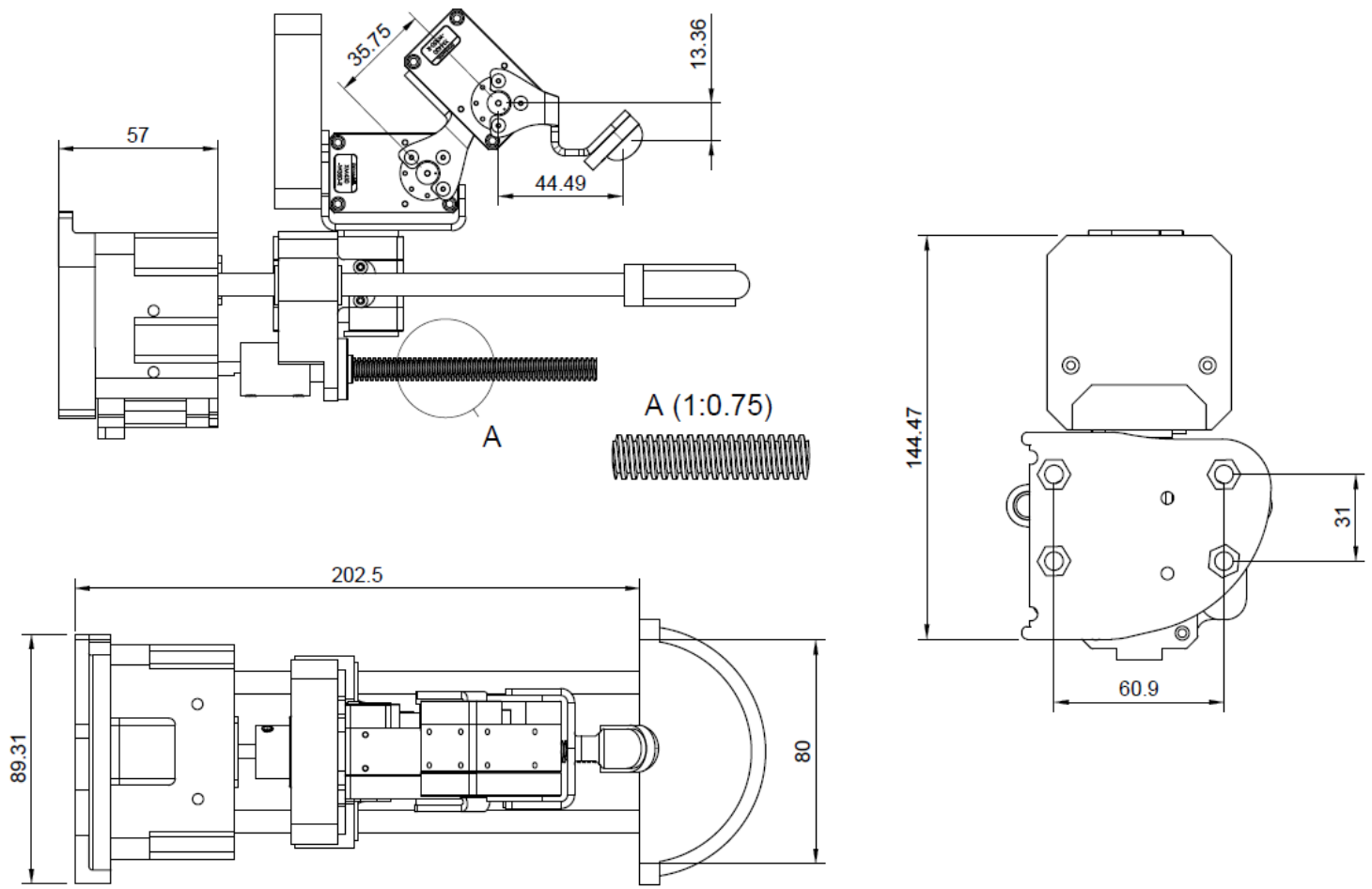
The final Lagrangian takes these components and converts them to torques and forces via Equations 4.29 or 4.30. The MATLAB code, which calculates the Lagrangian alongside $M(q)$, $C(q, \dot{q})$, and $G(q)$ is presented in Appendix B.0.4. The calculated expressions for actuator forces are then presented in Appendix B.0.5. Furthermore, the dynamics inform actuator choice and indicate that the chosen components are sufficient to move and control the proposed gripper during fabrication as Section 4.3.5 demonstrates.

4.3 System Integration and Assembly

The developed gripper's components, design and integration aspects evolved across several iterations after completion of the modelling procedure. Initially, the development process created prototype components using fused deposition modelling (FDM). However, the final mechanism used a combination of bent aluminium components, FDM parts, actuators, and sensors alongside custom PCBs, as Figure 4.3 depicts. In addition to this fabrication process, the manipulator required integration into a ROS network while communicating with custom hardware components. This section describes the implementation details in conjunction with information surrounding the integrated actuators. Finally, a discussion surrounding the ROS components used to interface the system for data-driven applications is presented.

Figure 4.3 displays the developed prototype, highlighting the sensors, actuators, links, fabricated parts, and chassis components while the gripper is attached to a Baxter robot. Figures 4.10 and 4.11 present a mechanical drawing and exploded assembly diagram of this prototype. The CAD program Fusion 360 provided the tools to develop the prototype and embed the various hardware components into a single mechanical system. The final prototype uses bent aluminium plates to construct the links connecting the rotational actuators. Aluminium is suitable for these components as the rotational actuators are the components forming grasps against the static plate, thus requiring physically robust and lightweight links while exerting torques and grasping forces. The remaining parts are FDM-printed components that comprise the chassis and connection mechanism to a robot arm. All hardware involved in the manipulator is attached to the chassis. The final MVP with the circuitry and sensors weighs approximately 1.4Kg.

As Figure 4.12 shows, the final gripper prototype was attached to a Baxter robot to evaluate data-driven approaches and external vision modules. A Python API (Application Programming Interface) using a ROS network enabled users to send desired position, velocity and torque commands to the robot manipulator. The API could also obtain the latest state of the manipulator, including the current configuration, sensor feedback, and metadata from the hardware.



Notes: Circuitry is omitted in this view
 Units are in mm

Figure 4.10: An orthogonal engineering drawing of the developed gripper.

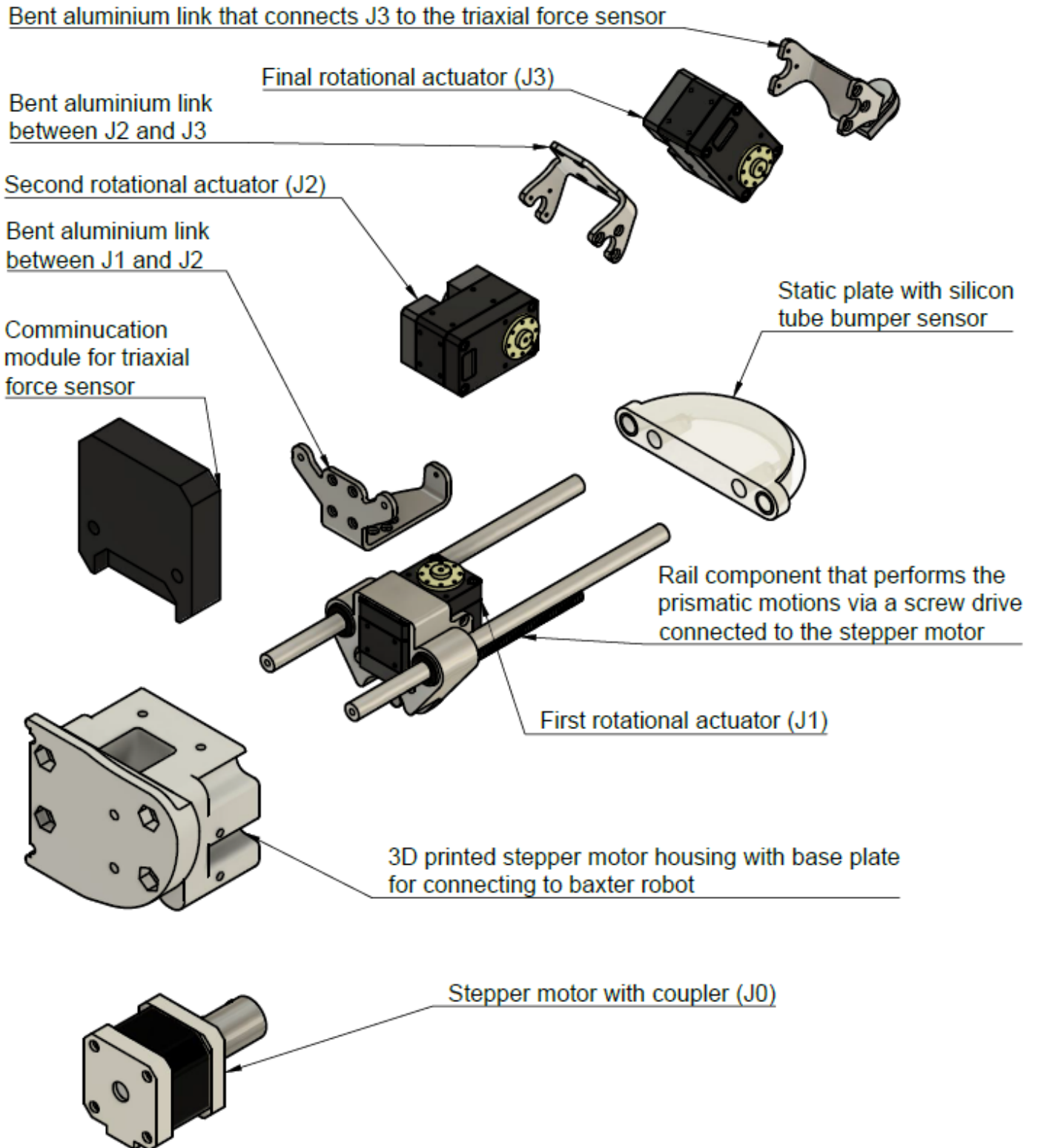


Figure 4.11: An exploded assembly diagram of the developed gripper.

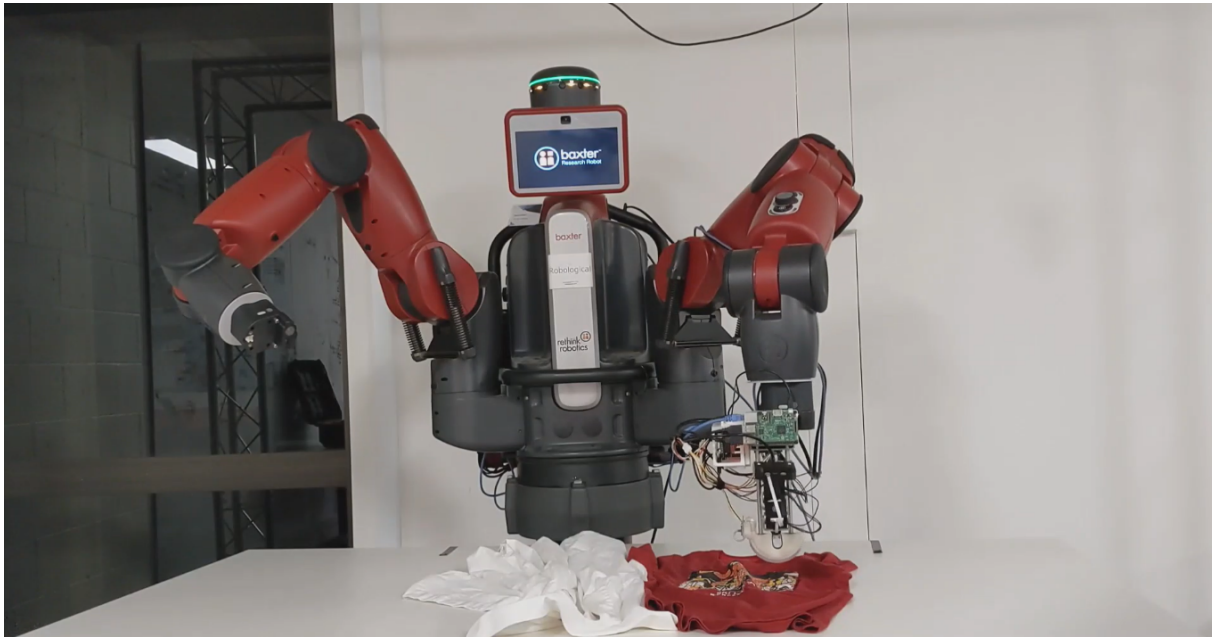


Figure 4.12: The final gripper prototype attached to a Baxter robot.

4.3.1 Actuator Components

Prismatic rail with a stepper servomotor

The first actuator chosen was the component that could provide motion to the prismatic joint (J_0). This initial prismatic joint slides the thumb appendage along the manipulator's chassis, enabling the device to alternate between the LEG and LFG configurations (Figure 4.1). Additionally, this motion allows the fingertip to reach beyond the static plate component and interact with the environment while performing EC grasping. A stepper motor coupled to a threaded rod with a lead screw achieves the desired linear motion. Two supplementary non-threaded rods hold the structure together for support. Figure 4.13 highlights this component with the chassis, the static plate surrounding the rail, and the threaded rod components attached to the stepper motor.

A stepper motor device was ideal for this component. Stepper motors are a type of actuator that can operate in ticks or steps. A series of coils called phases perform this functionality via currents that activate an electromagnet effect. Each time a set of coils are activated, the magnetised rotor aligns to the next step or tick. Such a mechanism enables an open-loop form of position control. Commercially available FDM printers use stepper motors for the precise control needed to move through three axes while printing CAD files. After evaluating several stepper motors through the iterative **Design Thinking** approach, the final prototype uses a NEMA-17 motor with a 38mm depth measurement (Appendix C.2 for details). Per specifications from the purchased device, each phase draws 1.7 A (*Amps*) at 2.8 V (*Volts*),

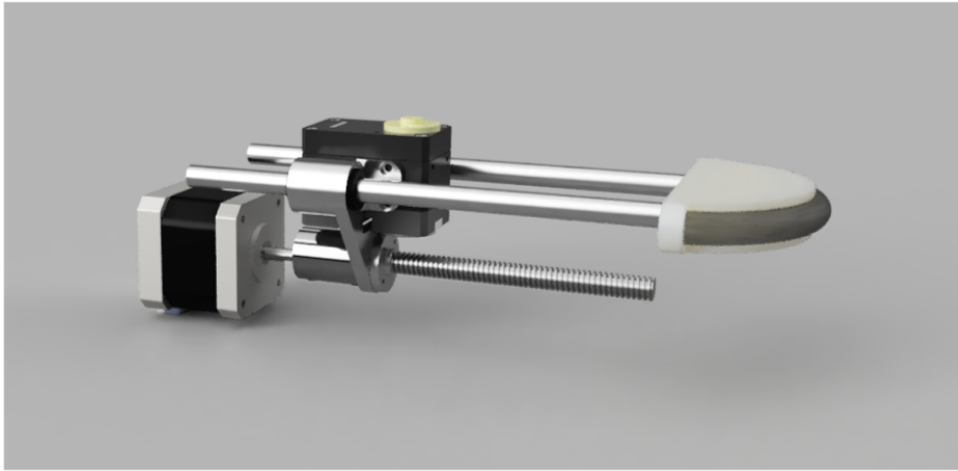


Figure 4.13: Rendering of the prismatic rail component coupled to the stepper motor ($J0$).

allowing for a holding torque of approximately $0.36Nm$ (*Newton-meters*). As deployment took place, the stepper drive set the phase limit to $1.32 A$.

The mechanism that performs the linear motion is the coupling of the stepper motor to a threaded rod. Thus one can model a relationship between the torque exerted by the stepper motor and the linear force applied by the actuator depending on the threaded rod parameters. A stainless steel T8 ACME 8mm threaded rod is used in this design, a common component that converts the rotation of a stepper motor to linear motion. The physical parameters of this component are a thread pitch (P) of $2mm$, a lead length of $8mm$, and an assumed friction coefficient of $0.25N$. Additionally, the ACME thread characteristics mean that a thread angle of 29 degrees, defined as α_{th} , is present in the thread. Finally, a variable of mean rod diameter d_m is present with a value of approximately $7.5mm$. d_m derives from the expression $d_m = d_o - \frac{P}{4}$, which approximates the depth of thread on the rod. Assuming the holding torque value given in the previous paragraph, Equation 4.39 estimates the maximum holding force of this prismatic joint. The resulting maximum holding force is equivalent to $146.5N$ using the values of these threaded rod parameters. Note that this expression assumes the maximum raising force using the holding torque parameters for this stepper actuator. Once moving, the exerted force may differ.

$$F_M = \frac{2T_R}{d_m} \left(\frac{\pi d_m - f_c l \sec(\alpha_{th})}{l + \pi f_c d_m \sec(\alpha_{th})} \right) \quad (4.39)$$

Dynamixel XM-430-350-R

The XM-430-350-R actuators are commercially available rotation servo-motors designed for robotic applications from the company Dynamixel. Several features surrounding these devices make them well

suiting to grasping applications. Firstly, they possess in-built position encoders that provide present position and velocity feedback details. Furthermore, metadata such as the temperature or voltage are available. Additionally, these devices possess built-in lower-level control systems targeting position, velocity and torque modalities. More specifically, these motors offer an operating mode referred to as *current-based position control* mode, an impedance controller that takes in desired rotational positions and a target current value that sets the maximum current the actuator moves with, thereby modulating the force the actuator applies. Such a feature provides the capabilities to modulate the grasp force, as the rotational actuators are the only moving components involved in the gripper formulating a grasp. Additionally, the actuators contain free rotation bearings, which enables links with bracket connectors to robustly attach to the actuator.

This actuator can also provide approximately $4.1Nm$ of stall torque at maximum. Per recommendations from the manufacturer, the continuous duty rating of Dynamixel actuators is approximately 20% of the maximum stall torque referenced. Therefore, operation of this gripper limits the impedance controller to only use torque ranges up to a fifth of the allowed maximum. Even with this constraint, the manipulator can apply grasp forces appropriate for fabric manipulation (Section 4.5). An evaluation was required where an XM-430-350-R is attached to an FDM-printed structure and loadcell. Stall torque values were calculated from loadcell measurements while the actuator was given various goal current

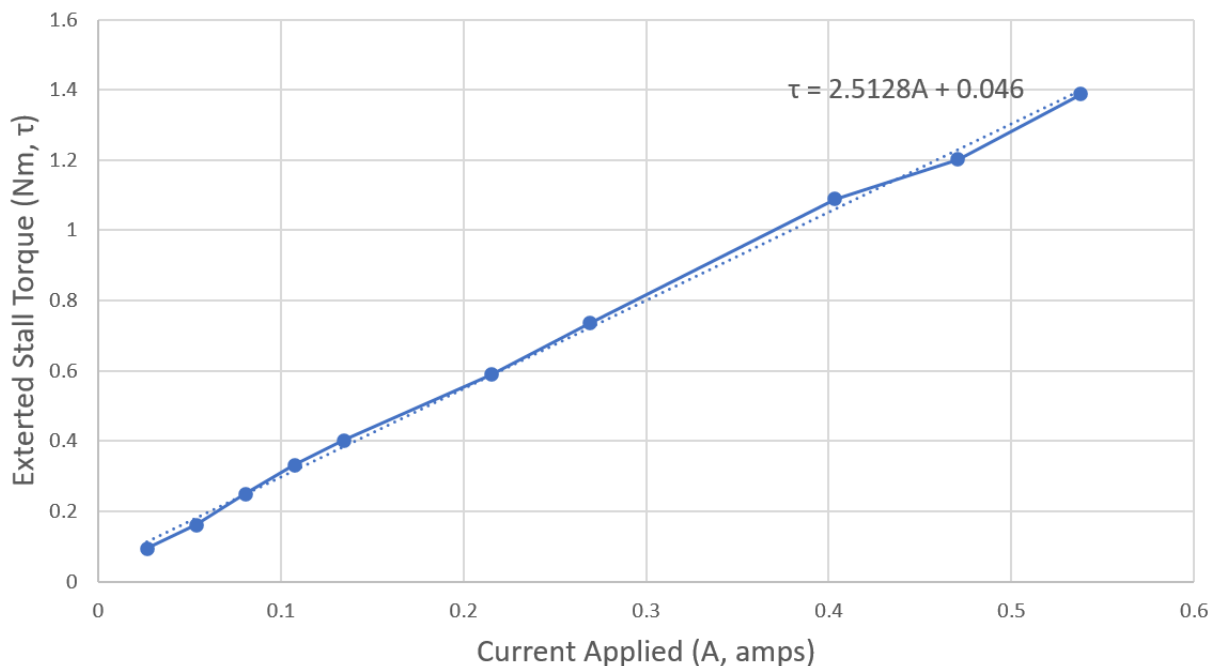


Figure 4.14: Measured stall torque readings of a Dynamixel XM-430-350-R actuator.

commands while pressing into the loadcell, thereby modulating the exerted torque. A linear relationship was confirmed, as shown in Figure 4.14. This figure also shows the maximum stall torque ($1.388Nm$) and goal current (0.538 amperes) applicable to the Dynamixel actuators while using them in the safety range to ensure continuous operation. These measurements occurred with an average voltage of 11.9V and a velocity limit of 46 RPM. These settings may influence the stall torque of these rotational actuators.

4.3.2 Electronic Components

In order to control the actuators and communicate the present state of the robot manipulator, a custom circuit board and communication protocol was required to maintain control of the robot and reliably communicate with the host ROS system. This section details the electronics and low-level serial communication that allows the manipulator to communicate across a ROS network, as depicted in Figure 4.15. The Dynamixel XM-430-350-R actuators use the RS-485 protocol to communicate with a serial interface. This setup uses a *max485* chip attached to the serial line (*TX/RX pins*) of a microcontroller. The microcontroller then uses a library available from the manufacturer to communicate with the servomotors across the serial line. Communicating with these actuators includes sending information such as the desired goal position and commanded current/torque, among other basic settings, and receiving information such as the present velocity, position, temperature and other meta-parameters.

Controlling the prismatic actuator, which uses a stepper motor, is slightly more complex. As the rotational actuators run low-level in-built controllers, integrating such devices requires a communication setup and an understanding of the actuator's capabilities. However, the stepper motor has no in-built controller. Therefore, for the prototype presented, a custom controller that accepts desired position and velocity commands was implemented to control the first joint of the manipulator. While details of the integrated stepper motor were present previously (Section 4.3.1) this section provides details about the circuitry and communication protocol implemented to transmit information from the microcontrollers to a higher level program.

An A4988 breakout circuit interfaced with an ATmega32U4-driven microcontroller for integration of the stepper motor with the electronics. The microcontroller then pulsed the NEMA-17 stepper motor depending on whether the actuator was at the target position, and a target velocity determined the pulsing speed. The ATmega32U4 device (known as the Qwiic Pro microcontroller) also acted as the controller, and processed commands via serial to set the target speed and position, while providing feedback on the present motion details. Commands given to the stepper were unsigned 16-bit values, which indicated the

linear position in tenths of a millimetre. A linear relationship converted the position information to pulse step values. The linear expression used a gradient of 2.4572 to convert the prismatic joint position value into a step value, informing the stepper motor rotation behaviour. One limitation of this approach is that controlling the stepper motor component in this manner remains an open-loop system. This description is apt as there is no mechanism to validate that steps have pulsed successfully, i.e. not skipping steps. As discussed further in this section, a laser measuring the linear displacement of $J0$ is present to partially address this issue.

The manipulator's primary control device was a Teensy 4.0 microcontroller, which sent serial data to the Dynamixels and the microcontroller handling the stepper behaviour. The Teensy also communicated using another serial line to the host PC, which controlled the manipulator's commands and feedback data. Additionally, several sensors which used the i2c bus interfaced with the Teensy microcontroller and provided further feedback information, including about the orientation of the manipulator via an inertial measurement unit (IMU), collisions with the environment using a pressure sensor, and the position of the prismatic actuator with a time-of-flight (ToF) laser. The IMU device was a breakout circuit connected to the i2c bus on the printed circuit board (PCB) attached to the hand. The pressure sensor evaluated readings in an airtight silicon tube at the end of the static plate appendage. This mechanism provided a signal to indicate if the manipulator had made contact with the environment surface. Finally, the manipulator used a ToF laser embedded in a position that enabled it to query the displacement of the rail component to verify $J0$'s position if required. However, the ToF laser takes over 100ms to collect a single reading. Therefore during operation, a ToF reading only occurs upon request, and pauses system serial communication while doing so. Such a solution is useful for calibrating the rail component, but cannot enable closed-loop operation for $J0$.

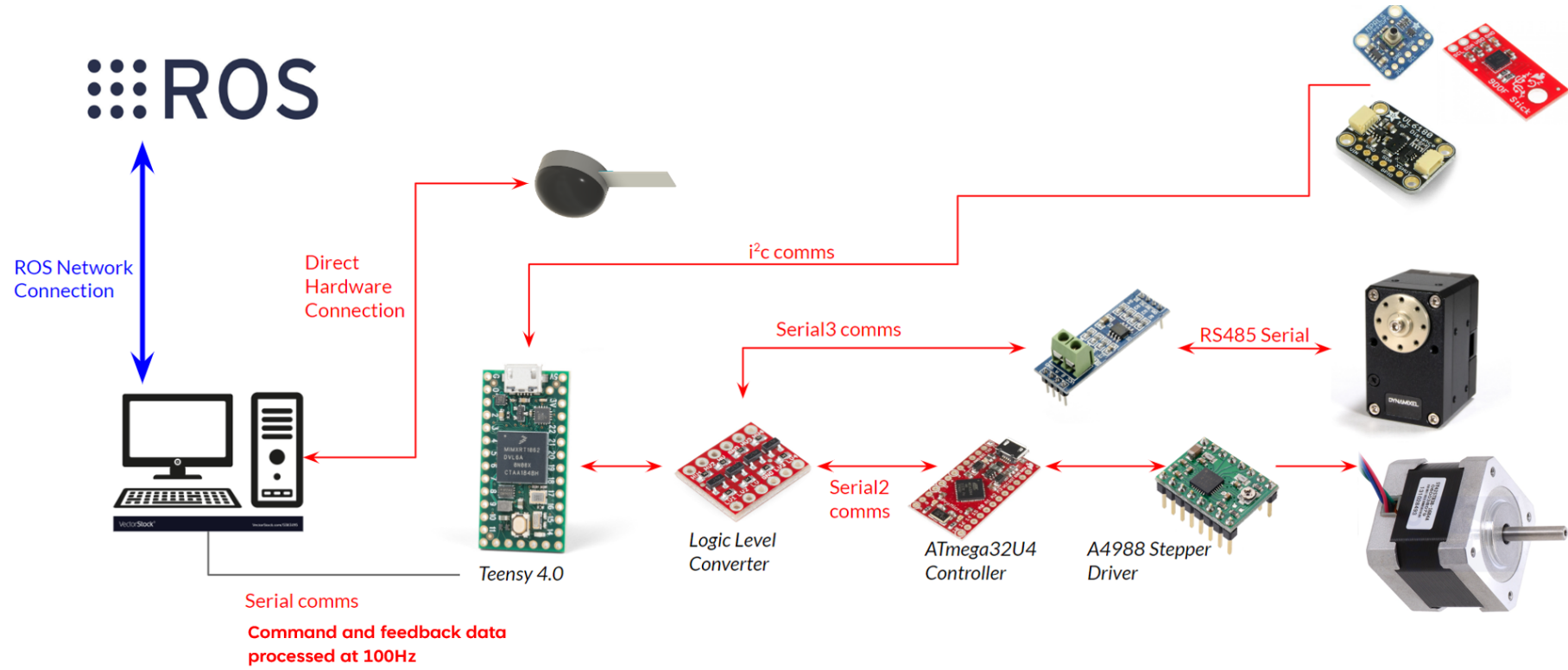


Figure 4.15: An overview of the electronic architecture.

Figure 4.3 indicates the positions of the sensors and actuator positions on the manipulator, while Figure 4.15 outlines the electronics. A list of components and their commercial sources is provided in Appendix C.2. In order to integrate the various components, a custom PCB provides the interface to communicate data, provide power and control the manipulator. A series of drop-down regulators are present on the PCB to control the power requirements of the manipulator. Firstly, a regulator that converts a 24V input power source to 12V is present, after which two subsequent regulators of 5V and 3.3V provide power for the microcontrollers, breakouts and i2c sensors. Finally, logic-level converters are also present to communicate between devices with different logic voltages. The circuit design uses two dual-layered PCBs using standard 2.54mm PCB headers to connect, these being located in a detachable 3D-printed case connected to the main chassis.

4.3.3 Software

Serial Communication

As previously indicated, serial communication plays a prominent role in the operation of this manipulator for communicating between microcontroller components and interfacing with the ROS network. A communication protocol that dynamically structures byte arrays provides an interface across the various hardware components. This protocol is necessary as data to command the actuators and receive feedback from the sensors uses a range of positive and negative integer values. However, these values do not exceed the range of 16-bit numbers; accordingly, the message protocol assigns two 8-bit integers (bytes) to each message, representing the destination and command. Table C.1 in Appendix C.3 describes the available destination and command options. For example, a message may define a destination of *J0* and a command of *TARGET-POSITION*. Two more bytes then define a 16-bit integer split by bit-shifting, representing the desired target position of *J0*. A byte array then embeds these messages with additional metadata, including checksum bytes and payload information, before sending the array across a serial line. Table 4.2 visualises this message protocol and data definitions that parsed manipulator status information.

Table 4.2: The byte message structure that communicates between the ROS host and microcontrollers.

		<i>{List of commands (L_c)}</i>				
Header	Payload	Destination	Command	Payload_upper (u)	Payload_lower (l)	Footer x2
0x00	<i>length(L_c)</i>	uint8 D	uint8 C	JOIN(uint8 u, uint8 l) (u <<8) l		0x00

ROS architecture

The final aspect of development discussed is the implementation of the ROS architecture, which connects the data from the gripper to a local network, enabling RL implementations and general python scripts to send commands and receive feedback. As described in Section 4.3.3, a serial line on the Teensy microcontroller sends byte array data which communicates with a ROS node that parses this information through to a ROS network. Figure 4.16 displays the information of a custom ROS message that encodes all the information of the gripper's state. For the actuators, present motion information such as the position and velocity are available. Furthermore, due to the in-built microcontroller and sensors of the rotational actuators, additional information, including the temperature, voltage, and alarm status, was available and utilised. Finally, target position and current (torque) commands were available for the rotational actuators, while the prismatic actuator used target position and velocity commands. The actuator commands are listed in Figure 4.16, which highlights these components in red text. Finally, the gripper state holds data from the triaxial force sensor and *i2c* sensors, except for the ToF laser, which embeds

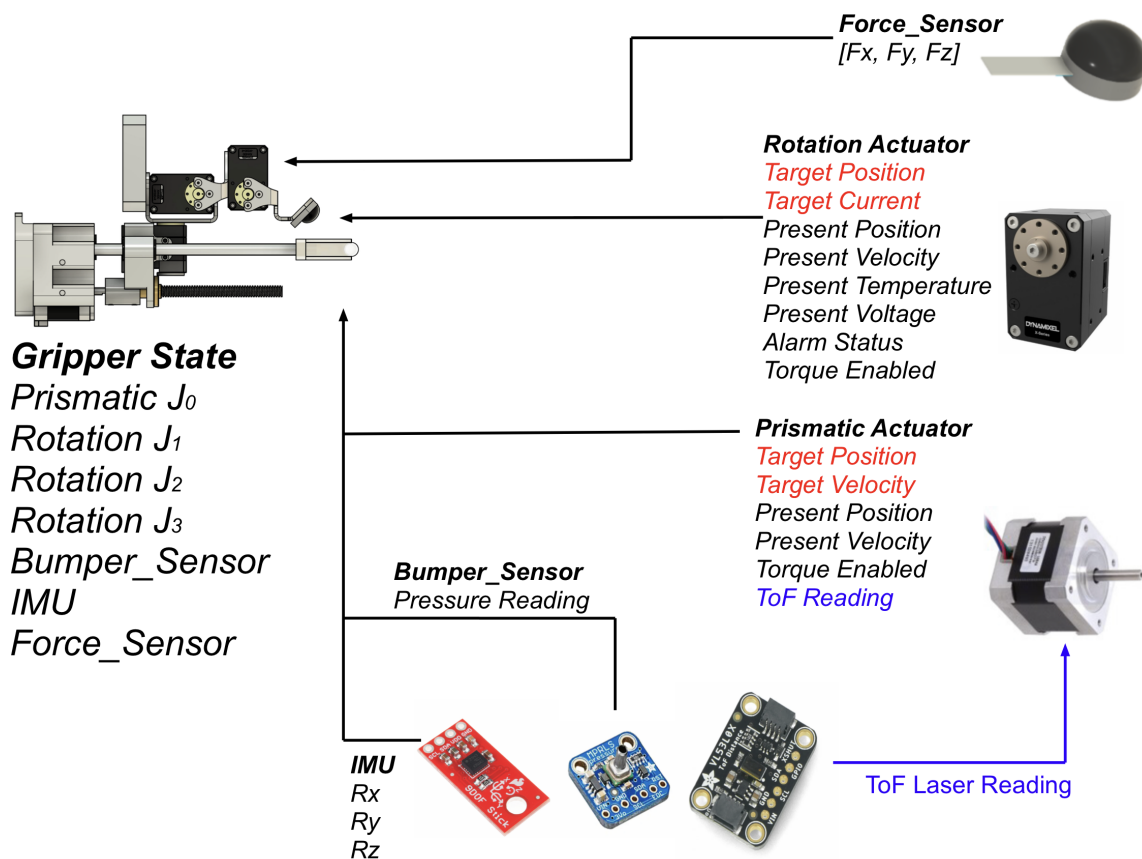


Figure 4.16: The custom ROS message structure for the gripper.

readings into the state of the prismatic actuator.

4.3.4 System Evaluation

Evaluating Operating Speed

Given the software details and hardware components discussed above, this section evaluates operation speeds and communication latency. The first investigation studied the duration of the gripper's operation loop and firmware modules, thereby informing operation speed alongside accommodations required to integrate various sensor and communication modules. Evaluating these temporal aspects of the firmware involved sending debug serial messages that record the duration of various operations in the firmware. The evaluation process collected 10000 duration values for three modules within the firmware, these modules being the operation loop, the Dynamixel interface, and the sensor interface. Table 4.3 presents the results and describes the modules analysed. The results indicate that obtaining gripper feedback from the sensors and actuators takes approximately $8ms$, while commanding the Dynamixel actuators averages around $2ms$. However, each operation loop took approximately $30ms$ to execute. The longer duration of the operation loop could be due to issues with inter-controller or ROS host communication modules. The final results indicate that the gripper can reliably operate at $30Hz$ for commanding the system and providing reliable feedback regarding the device's state.

Evaluating Serial Latency

In order to evaluate the communication latency during operation, a ping process occurs in which the serial ROS node on the host computer sends a request message to the microcontroller. Once received, the

Table 4.3: Microcontroller evaluation of the gripper.

	<i>Description</i>	<i>Mean (ms)</i>	<i>Mode (ms)</i>	<i>Median (ms)</i>	<i>Min (ms)</i>	<i>Max (ms)</i>	<i>Standard Deviation (ms)</i>
Control Loop Duration	The speed at which the control loop operates.	30.11	30	30	29	31	0.68
Dynamixel Interfacing Duration	The duration to command the Dynamixel actuators.	1.835	2	2	1	4	0.59
Sensor Interfacing Duration	The duration of executing the sensing modules and obtaining Dynamixel feedback.	7.95	8	8	7	8	0.21

microcontroller sends a response message to the ROS host. This evaluation records the duration of the process to gather an understanding of the serial communication latency. Similar to the evaluation in the previous section, this exercise runs 10000 pinging actions to estimate the system latency and the associated variance. This process occurs for communications to the Teensy microcontroller and the Qwiic Pro microprocessor that handles the prismatic joint operation. While simultaneously collecting this duration data, an additional process acquires the latency duration for the ROS network. This second evaluation process replicates the pinging action described above for both microcontrollers, but the duration measures the time a ROS node could send and receive a message from the hand. Such an evaluation measures the latency across the ROS network, which also requires the serial communication line to send data to the hardware.

Table 4.4 displays the results of these pinging evaluation processes. Per the results of Table 4.4, serial communication between the ROS host and both microcontrollers returned an average ping latency between $12ms$ and $14ms$. The evaluation of the Teensy microcontroller saw a significant increase in the mode latency value when compared to the mean for both the serial and ROS network ping evaluation. Such phenomena could be attributed to the increased processing load on the Teensy as opposed to the Qwiic Pro microcontroller, as the Teensy may have delayed responses while gathering sensor and actuator feedback while processing the primary operation loop. In contrast, the Qwiic Pro only reads serial data and controls the stepper actuator. Considering that this data refers to a pinging action that sends two

Table 4.4: Communication latency evaluation of the gripper.

	<i>Description</i>	<i>Mean (ms)</i>	<i>Mode (ms)</i>	<i>Median (ms)</i>	<i>Min (ms)</i>	<i>Max (ms)</i>	<i>Standard Deviation (ms)</i>
Teensy Serial Latency	The speed at which the Teensy communicates with the ROS serial node via a pinging serial request.	12.8	22	10	1	31	7.45
Teensy ROS Latency	The speed at which the Teensy can interface with a python program that commands the gripper across the ROS network.	29.99	40	28	6	60	7.87
Qwiic Serial Latency	The speed at which the Qwiic Pro communicates with the ROS serial node via a pinging serial request.	13.6	12	13	3	33	3.37
Qwiic ROS Latency	The speed at which the Qwiic Pro can interface with a python program that commands the gripper across the ROS network.	30.7	30	30	10	53	3.44

messages, one can infer that, on average, serial communication between the gripper and the local ROS node can communicate at 100Hz during operation. Furthermore, the pinging process averaged approximately 30ms for both microcontrollers for the evaluations that considered the ROS network. This level of responsiveness should remain acceptable to operate the gripper in data-driven applications.

4.3.5 Estimating Dynamics from CAD

Using a publicly available add-on³, Fusion 360 allows users to generate a URDF (Unified Robot Description Format). URDFs are XML (Extensible Markup Language) files that robotic simulations use to integrate models of mechanical structures. URDF files also use the physical properties of CAD software to estimate dynamic variables, including the inertia tensors and CoM parameters, relative to each component. However, the link transformation structure is mismatched from the kinematic parameters (Section 4.2). Figure 4.17 visualises the URDF while highlighting file coordinate frames, visualising this discrepancy. Compared to the kinematics presented in Figure 4.6, the URDF coordinate frames do not align with the traditional kinematics.

³<https://github.com/syuntoku14/fusion2urdf>

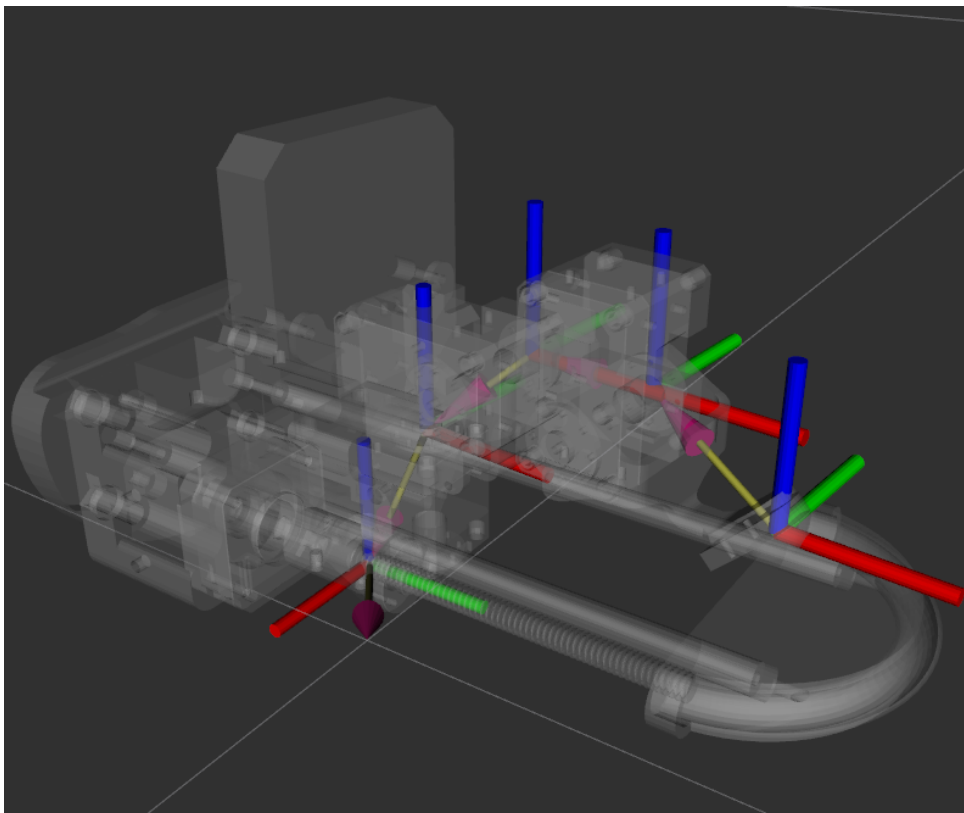


Figure 4.17: The gripper URDF model, visualised with RVIZ.

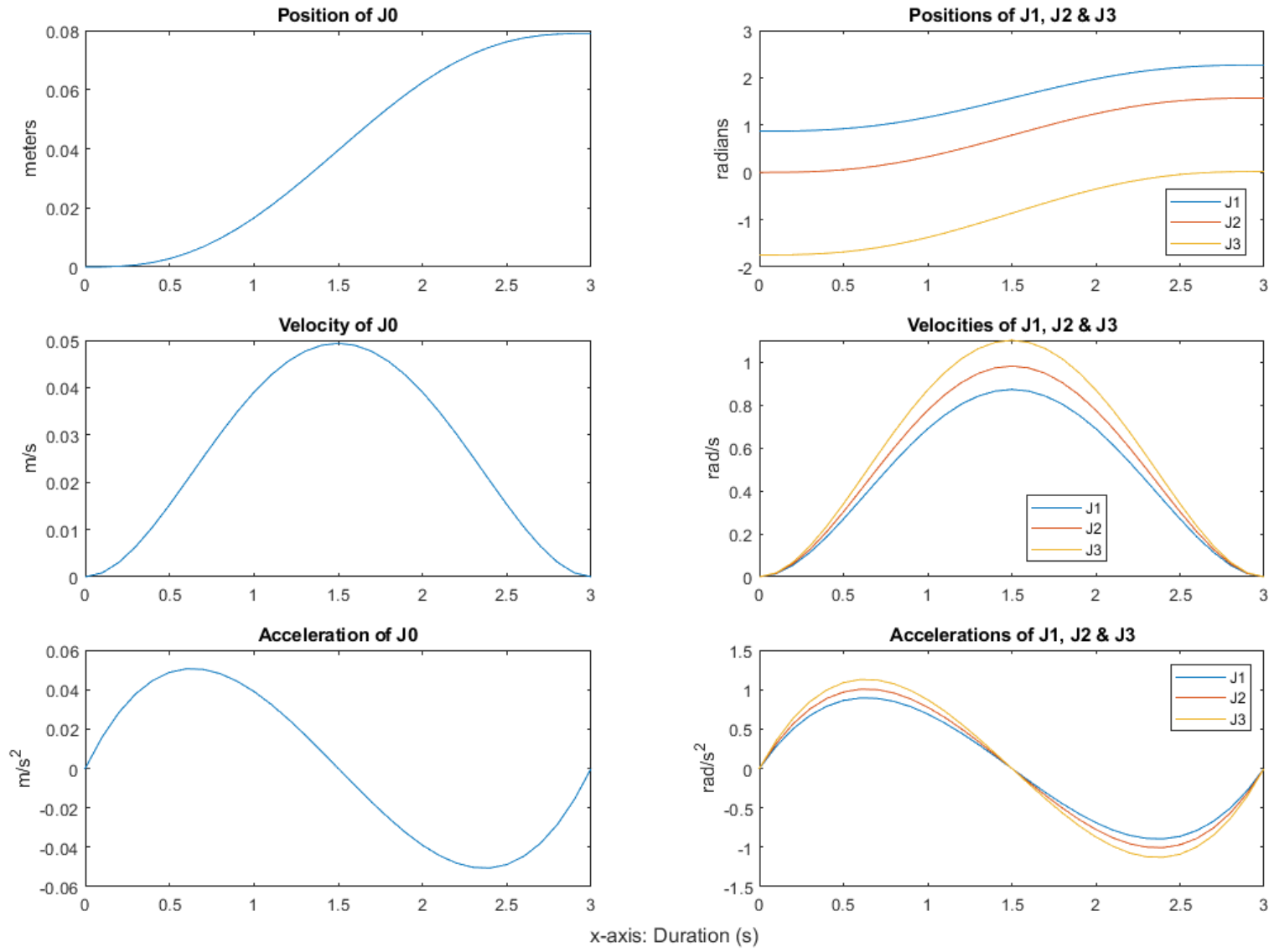


Figure 4.18: The example trajectory made to estimate the required actuator forces.

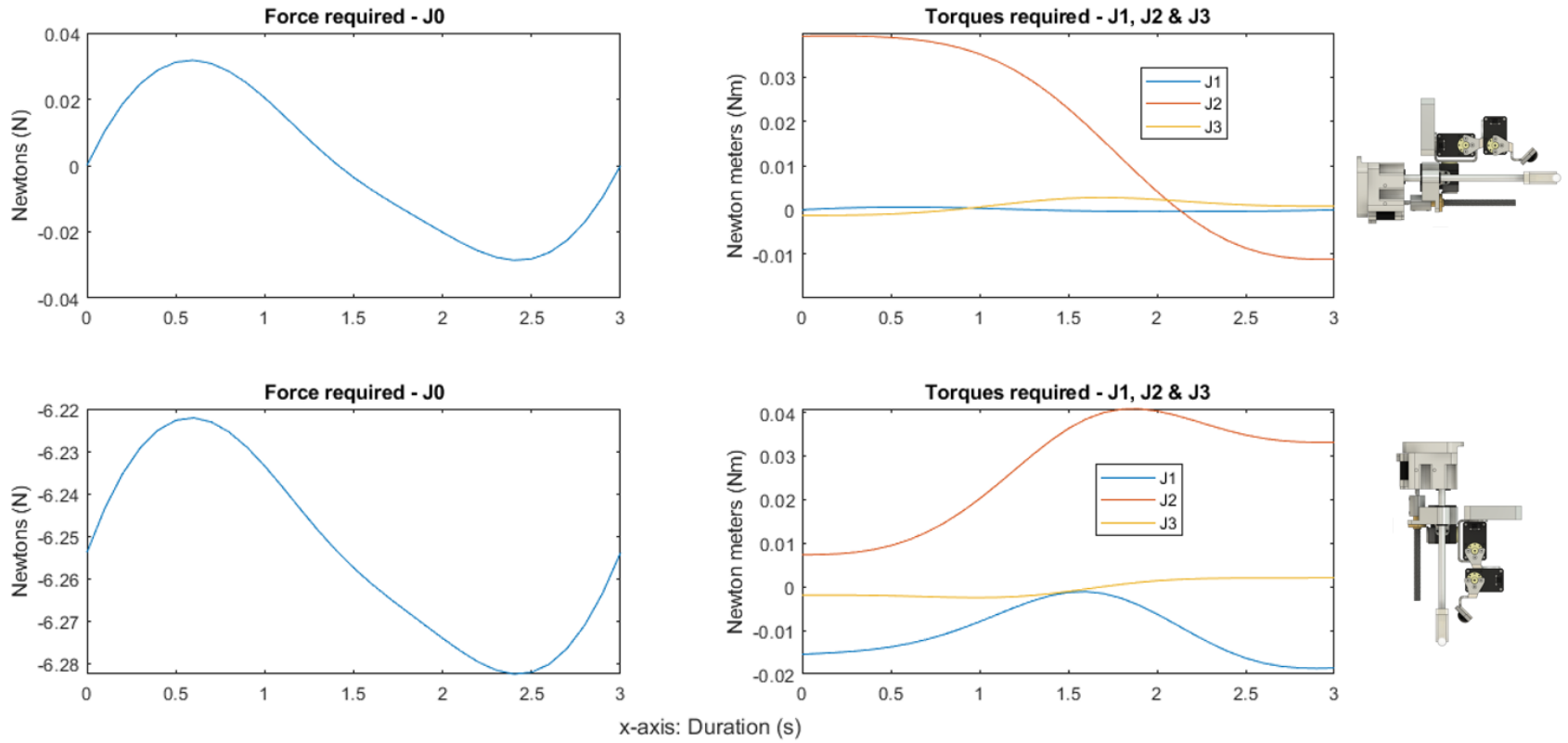


Figure 4.19: The forces required to execute the trajectory of Figure 4.18 under two orientations. Each row represents the gripper at a different orientation visualised on the right.

Table 4.5: The dynamics parameters of the URDF and derived variants for the modelling process.

Original URDF parameters	<i>Inertia Tensor ($kg \cdot m^2$)</i> [$xx,yy,zz,$ xy,xz,yz]	<i>CoM transformation (m)</i> [x, y, z]	<i>mass (kg)</i>
Link 0 (ul_0)	N/A (uses prismatic joint)	[0.0, 0.006, 0.026]	0.2032
Link 1 (ul_1)	[0.000217, 0.000241, 0.000187 1e-06, 6.7e-05, -1e-06]	[-0.013, 0.0, 0.034]	0.3135
Link 2 (ul_2)	[2.9-e05, 3e-05, 2.1e-05, 0.0, -1e-06, 0.0]	[0.034, -0.019, 0.011]	0.1085
Link 3 (ul_3)	[3e-06, 3e-06, 4e-06, 0.0, 1e-06, 0.0]	[0.022, -0.019, -0.007]	0.0123
DH model dynamic parameters			
Link 0 (hl_0)	*	[-0.0001, -0.0161, -0.0235]	*
Link 1 (hl_1)	[2.17e-04, 1.87e-04, 2.41e-04, -6.70e-05, 1.00e-06, 1.00e-06]	[-0.0330, 0.0134, 0]	*
Link 2 (hl_2)	[2.90e-05, 2.10e-05, 3.00e-05, 1.00e-06, 0, 0]	[-0.0017, 0.0113, 0.0010]	*
Link 3 (hl_3)	*	[-0.0225, 0.0000, -0.0070]	*

The methods for estimating the DH-based dynamics parameters from the URDF are in Appendix C.4.

Estimating the dynamics parameters of the modelling process from the URDF file requires further considerations. First, the URDF from the CAD script uses the world coordinate of the CAD software to generate the joint transformations, hence the discrepancy between the traditional kinematics and the generated URDF. Additionally, the URDF format estimates the local constant CoM vector from the joint actuating the link, whereas the convention used in the modelling process of Section 4.2 takes the inertia tensor and CoM parameters from the end of a DH row, i.e. the final transform of the joint link pair. Finally, as mentioned in Section 4.3.1, the Dynamixel actuators can attach bracket links via a free-drive bearing aligned to the actuator horn. The final two links connect in such a configuration. Therefore a simplified assumption is made that the actuation point is between the driving horn and the free-drive bearing. Conversely, in the URDF of Figure 4.17, the actuation point is directly upon the free-drive bearing components. With these considerations in mind, Table 4.5 presents both the URDF dynamics parameters and the converted values for the DH-based modelling approach from Section 4.2.

Finally, using these dynamics parameters and the robotics toolbox of Corke [Corke 17], this section creates an example trajectory to observe the required actuator forces while moving the gripper. The RNE (Recursive Newton-Euler) algorithm estimates these required forces. When calculating the Lagrangian in Appendix B.0.4, the formulated Lagrangian was symbolically equivalent to the RNE output of the

toolbox. The same trajectory is evaluated twice, with the gripper in a horizontal base position and another orienting the gripper downward. Figure 4.18 displays this example trajectory's positions, velocities and accelerations, while Figure 4.19 displays the required forces. The motion forces required are well within the capabilities of the actuators described in Section 4.3.1. For example, Figure 4.19 indicates that the trajectory requires torques between $-0.02Nm$ and $0.04Nm$ to perform the trajectory with these dynamics parameters. The Dynamixel XM-430-250-R actuators used in the final MVP are capable of exerting $1.388Nm$ of stall torque at maximum during gripper operation. Similarly, the trajectory requires forces between $0N$ and $6N$ during operation for the linear rail, which the chosen stepper motor and rail mechanism can exert per Section 4.3.1.

4.4 Environmentally Constrained Grasping

A vital aspect of this gripper is the ability to grasp flattened clothing from various wrist orientations using *biomimetic* grasping motions, also known as *arbitrary* grasping per definitions from Chapter 3. The gripper's static plate is semi-cylindrical with uniformly rounded edges, as Figure 4.20 visualises. Using geometry and spatial mathematical operations, one can calculate a transform that estimates the contact point of the static plate colliding with an environmental surface as visualised in Figure 4.21 using the toolbox of Corke [Corke 96]. Assuming the DH parameters of Table 4.1, Equation 4.40 performs this calculation using additional constant variables, including the displacement of the static plate along the respective axis of frame w (x_d , y_d and z_d) and the size of the static plate determined with two radii variables y_r and z_r , Where y_r refers to the radius of the static plate along the xy plane, and z_r referring to the radius of the rounded edge. Additionally, two input parameters, y_{rot} and z_{rot} , indicate the orientation of the gripper relative to the environmental surface. These input parameters are limited to values between -1.39626 and 1.39626 radians⁴. The final function of mp (map-point) is in Equation 4.40, which calculates the contact point positional transformation.

$$mp(y_{rot}, z_{rot}) = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} x_d + \cos(z_{rot})y_r - (z_r - z_r \cos(y_{rot})) \\ y_d + y_r \sin(z_{rot}) \\ z_d + z_r \sin(y_{rot}) \end{bmatrix} \quad (4.40)$$

$$C^P = T(mp(y_{rot}, z_{rot}))R_z(z_{rot})R_y(y_{rot}) \quad (4.41)$$

⁴ $1.39626_{rad} = 80 \text{ deg}$

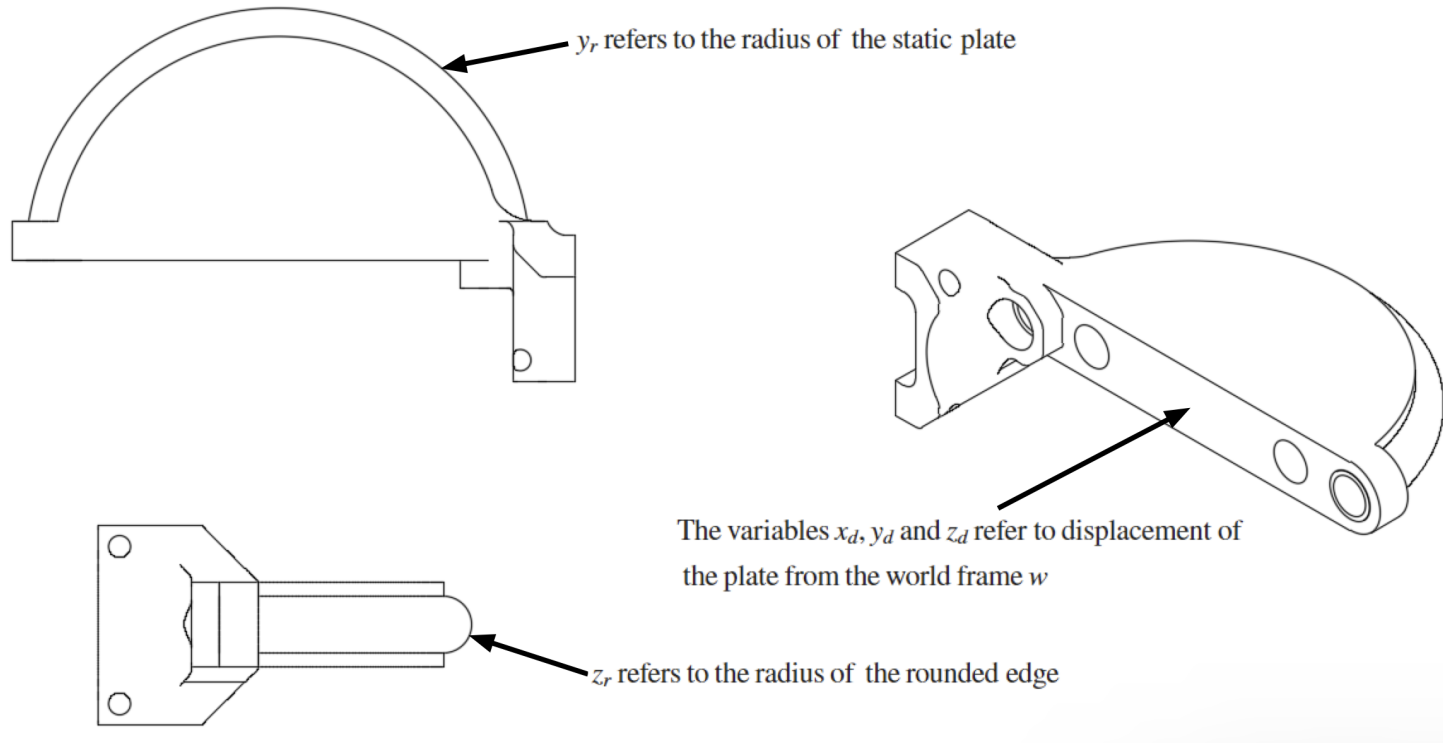


Figure 4.20: A visualisation of the static plate component demonstrating the variables for estimating a surface's position when the gripper makes contact.

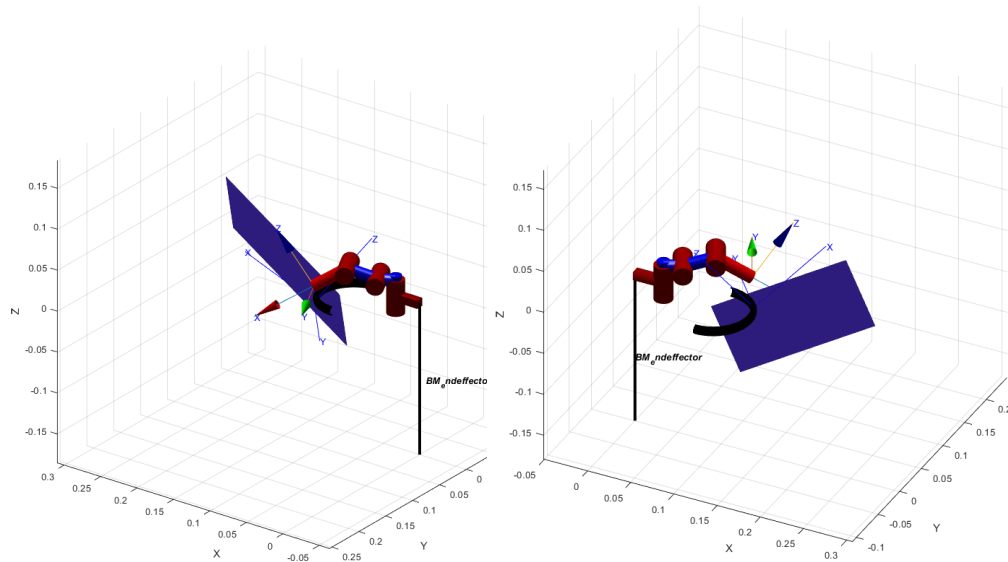


Figure 4.21: Calculation of level surface contact visualised with a y_{rot} and z_{rot} value of 40° .

Beyond calculating the position of contact, one could also estimate the location of a table surface using the transformation of C^p (Equation 4.41), as visualised in Figure 4.21. However, such a calculation assumes that the transformation has accurately estimated a perfectly level table and provides no further information about the friction or required forces to drag along the environment surface. While such a solution remains incomplete for informing a pipeline of grasping behaviour, the unique features of the static plate are still present. Additionally, when applying data-driven approaches to learn environmentally constrained grasping, the state of the gripper includes the orientation to inform grasping behaviour. In order to demonstrate the capability of *arbitrary* grasping, a publication which demonstrates an early-stage design of this gripper [Hinwood 20], demonstrates the gripper using the static plate to pin and

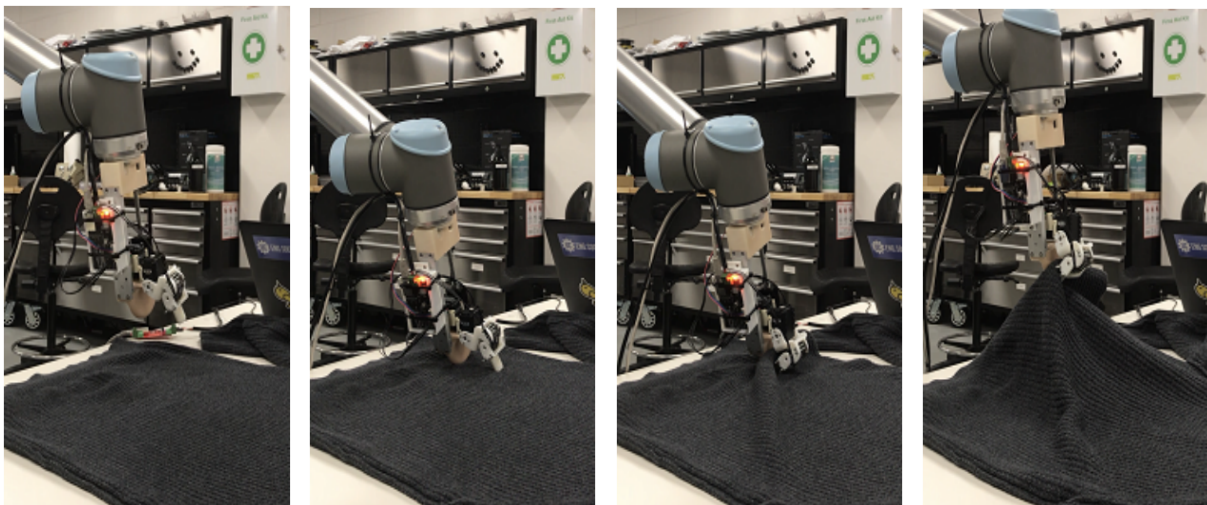


Figure 4.22: The initial EC grasping evaluation using pre-programmed trajectories [Hinwood 20].

grasp fabric with pre-programmed trajectories from a range of wrist angles. This design used position-velocity-controlled actuators to execute grasping behaviour on a flattened jumper. In 90 grasp attempts only a single failure occurred, this being when a double fold formed in the fabric and the garment slid out of the grasp while the arm was moving upwards. Figure 4.22 shows this preliminary evaluation during the gripper's early development.

4.5 Evaluation of Grasping Characteristics

This section characterises the grasp strength of the gripper. Evaluating the device's grasp force capabilities follows two processes. Firstly, the sensor Jacobian $J_s(q)$ mentioned in Section 4.2 and properties of the Dynamixel actuators from Section 4.3.1 estimate the expected grasp strength the gripper is capable of exerting. A loadcell placed on the static plate component measures the grasping forces exerted by the hardware device at various torques and gripper configurations. The readings from the loadcell are then compared to the expected values generated by $J_s(q)$. A loadcell is used to estimate the grasp force rather than the triaxial force sensor because the latter's exact centre point is difficult to orient directly onto the plate at various kinematic configurations. As previously mentioned, the triaxial force sensor experiences a degradation in accuracy as the point-of-contact moves away from its centre-point. Following this grasp force evaluation, an experiment similar to that of Donaire *et al.* [Donaire 20] takes place in which the gripper holds a range of garments at varying positions to demonstrate the maximum payload the gripper is capable of grasping.

4.5.1 Grasp Strength Validation

Forming a grasp with the gripper involves taking a pose that places the triaxial force sensor against the static plate, with deformable material held by this *point-to-plane* configuration. The impedance controller of the rotational actuators can then use a combination of target position and torque (current) values to ensure that the grasp is exerting a force against the static plate. The sensor Jacobian $J_s(q)$ can calculate the exerted forces and a moment (about the y -axis) in the world frame w , as shown in the kinostatic analysis of Section 4.2. Due to the DH parameters defined in Table 4.1 and the position of the static plate (placed along the xy plane of w , see Figure 4.21), estimating the grasping force with $J_s(q)$ becomes a matter of estimating the force applied along the z -axis of frame w . Therefore, assuming that the gripper is in a pose that collides with the static plate, the grasp force is calculable by taking the transposed inverse

of the sensor jacobian $(J_s(q)^T)^{-1}$, multiplying it by the actuator forces, and observing the force along the z-axis of w . The expression to estimate the grasp force is shown in Equation 4.42. The direction of the estimated force along the z-axis must also be negative as the sensor presses into the static plate in a downward direction relative to the z-axis of w .

$$\begin{bmatrix} f_x \\ f_y \\ f_z \\ m_y \end{bmatrix} = (J_s(q)^T)^{-1} \begin{bmatrix} \tau_0 \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad \begin{array}{l} \text{The highlighted value of } f_z \text{ represents the grasping force} \\ \text{against the static plate. A calculated grasping force will al-} \\ \text{ways be negative due to the direction the TCP presses into} \\ \text{the static plate relative to } w. \end{array} \quad (4.42)$$

As grasp formation only uses the final two actuators of the system, this evaluation operates under the assumption that both J_0 and J_1 are stationary and not exerting any force while estimating the grasp strength. A range of torque commands under two grasping configurations have been measured on the gripper and measured with an attached loadcell (Figure 4.23). Table B.1 in Appendix B.0.6 displays these readings alongside the stall torque exerted by J_2 and J_3 , calculated with $J_s(q)$. When collecting

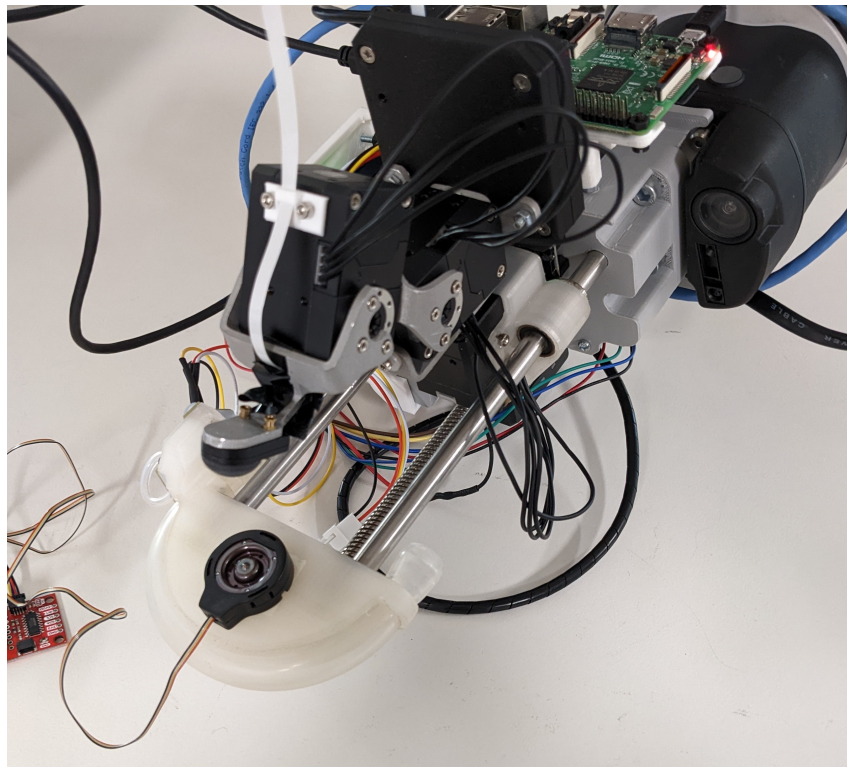


Figure 4.23: The fabricated gripper with a loadcell attached to the plate.

these loadcell measurements, $J2$ and $J3$ used the target pose values and goal torque commands shown in Table B.1. When forming a grasp, the stall torque exerted by $J2$ will always be greater than $J3$ due to the position of the static plate relative to both $J2$ and $J3$. Therefore, when using $J_s(q)$ to estimate the actuator torques from the loadcell measurements, this evaluation expects to observe that the commanded torque of $J2$ closely reflects the initial stall torque measurements from Figure 4.14. Figure 4.24 displays the Jacobian-estimated torque exerted by $J2$ while in the LEG configuration, while Figure 4.25 visualises the same information while the gripper is in the LFG configuration. Both figures compare the Jacobian-estimated torque of $J2$ to the stall torque estimates of Figure 4.14 while grasping.

Figures 4.24 and 4.25 show that the Jacobian-estimated stall torque readings exerted by $J2$ fell below the actual stall torque estimates in Figure 4.14. This difference could be due to one or more aspects of this evaluation. Firstly, a small FDM cap was placed on the loadcell sensor to ensure that the sensor was safely pressing into the loadcell. Such a setup may have impacted grasp force readings. Secondly, the sensor's position is taken relative to the sensing coordinate frame (A_s from Section 4.2.1) of the sensor, which influences calculations made with $J_s(q)$. Furthermore, A_s remains beneath the rubber overlay, so that damping caused by the rubber may have impacted force readings. In addition, a position on the rubber's body would have been in contact with the loadcell, not the position of A_s . A small amount of mechanical backlash was also present at the connection point between the servo horn of $J1$ and the aluminium link that connects $J1$ and $J2$, which may have also impacted loadcell readings. Finally, in certain circumstances, the target position of $J2$ was within 5 degrees of the resultant position while grasping (Table B.1), and as the difference between the target position and position of $J2$ fell, the position gain variable of the impedance controller used by the Dynamixel actuators may have further impacted loadcell measurements.

The grasp force readings from Table B.1 indicate that the gripper can grasp at a maximum of approximately $17.25N$ in the LEG configuration and $29N$ in the LFG configuration. As discussed at the beginning of this chapter, Le *et al.* [Le 13] recommend a maximum grasping strength of $30N$ for generalised garment handling. Therefore, this evaluation demonstrates that the developed gripper broadly meets the grasp strength requirements for such a task while retaining the capability to modulate applied grasp forces. However, this evaluation highlighted some limitations in the grasp modelling process when compared to the hardware prototype. Expanded modelling practices using a refined manipulator design with further experimentation should yield a more accurate system model.

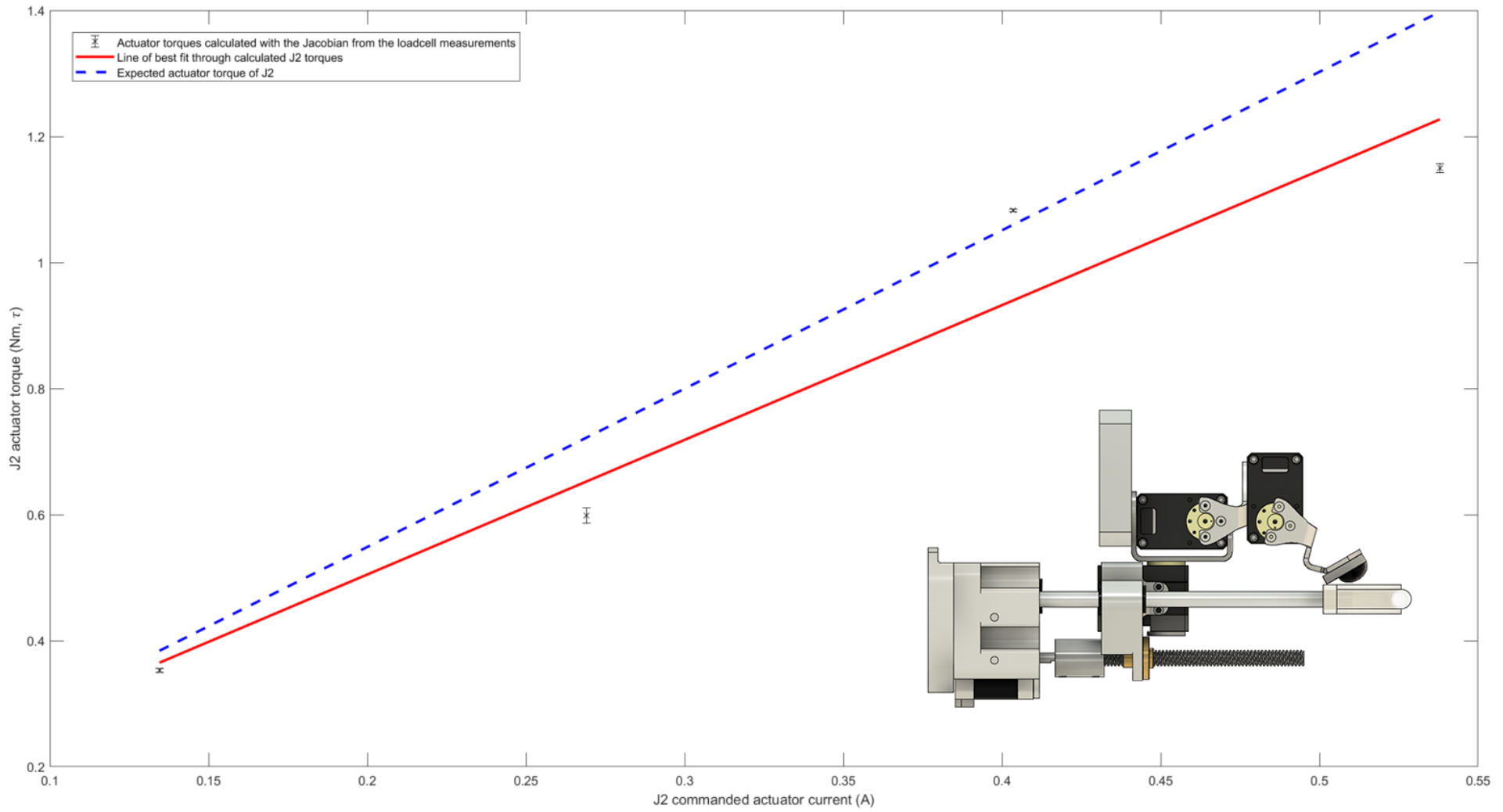


Figure 4.24: Comparison of expected torque running through *J2* while in the LEG configuration (visualised), compared to stall torque estimates of Figure 4.14.

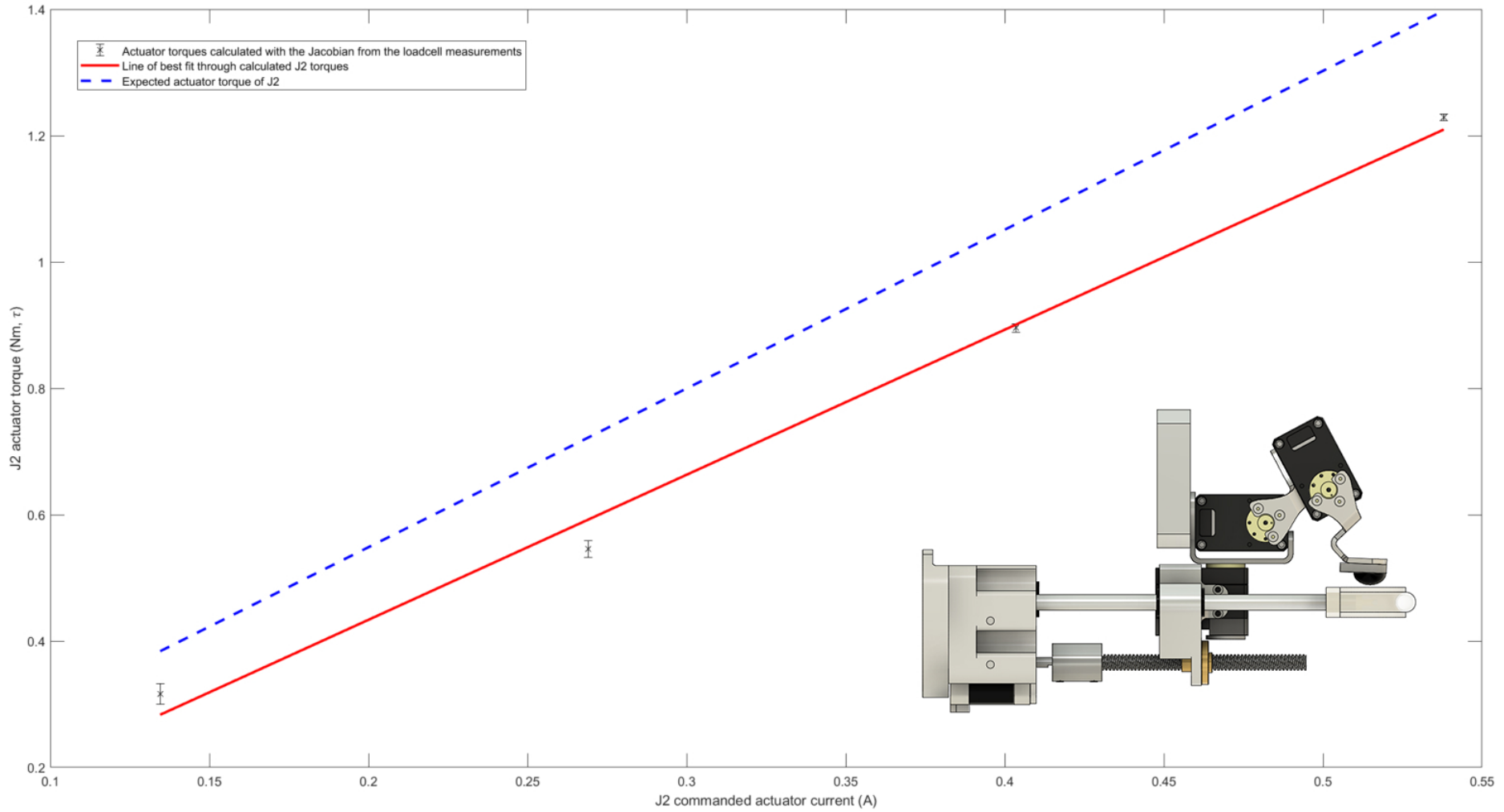


Figure 4.25: Comparison of expected torque running through $J2$ while in the LFG configuration (visualised), compared to stall torque estimates of Figure 4.14.

4.5.2 Holding Garments

In order to evaluate the functionality of the gripper while holding garments, another validation process was conducted to demonstrate the gripper holding various materials with additional weights to prove it is capable of holding a diverse range of garment payloads. Inspired by the experiment of Donaire *et al.* [Donaire 20], the gripper holds clothing for 20 seconds. An earlier prototype of the gripper (Figure 4.4(b)) failed to hold a garment weighing above 800g in a LEG pose, more specifically, a jumper object



Figure 4.26: The gripper holding various garments.

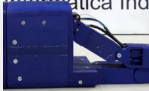








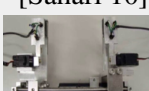
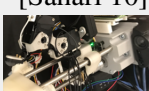
with an additional 300g. The set of garments evaluated in this trial included a jumper, neck-tie, jean shorts, scarf and t-shirt, as shown in Figure 4.26. While exerting the maximum available torque command, a Dynamixel goal current of 200, the final gripper prototype could hold all garments regardless of the kinematic pose being in the LEG or LFG configuration. A maximum additional weight of 550g was applied to each garment to give a maximum weight of 1050g (while holding the jumper) and the gripper was still capable of holding all target objects.

4.6 Concluding Remarks

The gripper formulated in Chapter 3 has been modelled, fabricated and evaluated as described in this chapter. This research has resulted in a mechanical system that uses existing serial-link modelling processes inspired by the approaches of Corke [Corke 17] and Yoshikawa [Yoshikawa 90]. Additionally, the device is capable of grasp force modulation and appropriate grasping forces needed for generalised fabric manipulation. This device also partially addresses the previously mentioned gap of being able to perform environmentally constrained grasps from various wrist orientations. These aspects contribute to the distinctive nature of this generalised pick-and-place fabric sorting solution. To compare the characteristics of this device with those of previously existing solutions, Table 4.6 compares grippers discussed in Chapter 3 with the fabricated device of this chapter.

Some aspects of this gripper require further exploration. For example, the device remains capable of haptic exploration, performing rubbing motions against the static plate. This aspect, alongside slippage detection and reaction, is beyond the scope of this thesis and will be further explored. A possible avenue to explore in this regard would be a replication of the haptic exploration setup of Drigalski *et al.* [Von Drigalski 17b], who enabled a gripper to distinguish a range of deformable materials with a machine-learning approach that interpreted rubbing data from two triaxial force sensors. In addition, a greater understanding of the grasp force and the rotational actuators' impedance controllers could support future skills embedded in this device. While the gripper in its present form has some limitations, it fulfils the outlined requirements by presenting a serial-link gripper that retains the capabilities for pick-and-place sorting of textile waste. Chapter 5 builds upon the formulation and development of the gripper by exploring data-driven approaches to environmentally constrained grasping of fabric.

Table 4.6: A comparison of the grippers discussed in Chapter 3 with the fabricated gripper.

	Grasp force range	Grasp force modulation	Grasps leveraging environment	Tendon driven components	Compliant joints
 [Donaire 20]	N/A	No	Yes	No	Yes
 [Le 13]	0-40N	Yes	Yes	No	Yes
 [Koustoumpardis 14]	N/A	No	Yes	Yes	Yes
 [Von Drigalski 17b]	4.7N	No	No	Yes	Yes
 [Koustoumpardis 17]	N/A	No	Yes	No	No
 [Ono 01]	N/A	No	Yes	No	No
 [Shibata 08]	N/A	No	Yes	No	No
 [Sahari 10]	5.47N	No	No	No	No
 [Sahari 10]	0.93N	No	No	No	No
 [Sahari 10]	3.3N	No	No	No	No
 <i>Fabricated Gripper</i>	0-29N	Yes	Yes	No	No

Chapter 5

Novel Motor Control Skills for Grasping with Environmental Constraints

Chapters 3 and 4 used the **Design Thinking** framework augmented with human-centric observations to formulate and develop a unique gripper for sorting textile waste. While the presented device can control actuator behaviour and interface system feedback, it lacks a higher level of ‘intelligence’ to execute dexterous behaviour. The gripper’s desired skill set (Section 3.6) includes the capability to perform arbitrary grasping, i.e. the ability to grasp flattened materials with environmentally constrained (EC) grasping from a range of wrist orientations. The ability to leverage the environment while grasping makes humans powerful manipulators of the world [Eppner 15]. Previous manipulators targeting fabric manipulation explore EC grasping with compliant joints and pre-programmed trajectories [Le 13, Donaire 20, Shibata 09, Koustoumpardis 14]. Unlike these devices, the gripper of Chapter 4 is a single serial-chain rigid body system, and pre-programmed trajectories are insufficient to handle the diverse scenarios an arbitrary grasping setting introduces. For example, a pre-programmed trajectory cannot adapt to a significant change in wrist orientation and may fail to execute the grasping motion. In addition, leveraging the environment while grasping fabric remains a complex collision-rich interaction where parameters such as the applied wrench, friction coefficient, and material properties influence grasp completion. Reinforcement learning (RL) has seen success in learning complex physical interactions for robots [Elguea-Aguinaco 23], where a neural network (NN) provides commands to system actuators that inform intelligent behaviour at a higher level [Ahn 20, Peng 20, Zhu 22]. Using these data-driven approaches to execute EC grasping remains an unexplored research avenue that this chapter investigates by asking: *Can reinforcement learning algorithms execute EC grasping by learning from a reward schema inspired by human-centric*

behaviour?

5.1 Introduction

5.1.1 Environmentally Constrained Grasping

Section 2.1.3 discusses the importance of grasps that exploit the environment, and the role they play in combating uncertainty and improving grasp robustness. Using RL to navigate contact-rich manipulation is a common approach [Suomalainen 22, Elguea-Aguinaco 23]. Given the status quo, this chapter approaches the challenge of learning the appropriate motions to execute arbitrary grasping. Building upon the human-inspired manipulation theme present throughout Chapters 3 and 4, this chapter takes observed human behaviour from the literature, specifically from Eppner *et al.* [Eppner 15], to develop a data-driven approach that allows the gripper learn the skills of arbitrary grasping. Such a task involves developing a reward schema that encourages RL algorithms to command the gripper of Chapter 4 to perform a biomimetic grasping motion that drags along the environment before closing into a grasp. As

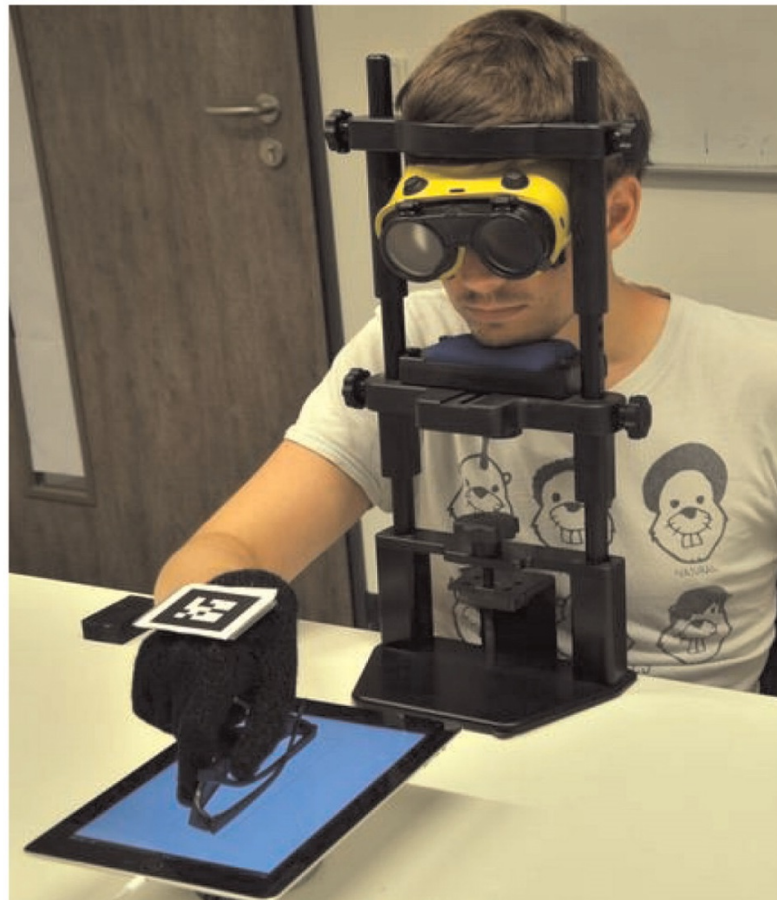


Figure 5.1: A participant in the user study conducted by Eppner *et al.* [Eppner 15].

humans consistently leverage the environment to improve grasp success [Puhlmann 16, Kazemi 14, Sarantopoulos 18, Della Santina 17], the reward structure to encourage EC grasping uses the human-centric conclusions drawn by Eppner *et al.* [Eppner 15]. They conducted a small human-centric survey in which five participants grasped objects from a flat surface, sometimes with visual impairments as shown in Figure 5.1.

This study found that participants would rely on environmentally exploitative grasps more often when visually impaired, indicating that such a skill can counteract uncertainty during manipulation. Furthermore, grasps that leverage the environment naturally saw an increase in interaction with the environment, inferring that a greater distance travelled along the environment occurred while moving into a grasp. These observations by Eppner *et al.* [Eppner 15] drive the formulation of the reward function developed in Section 5.2.2. When developing the simulated learning environment and reward structure, the distance travelled while in contact with the environment becomes part of the observation vector used to inform dexterous manipulation. This notion of distance travelled along the environment also contributes to the formulation of a reward function that encourages the gripper to perform EC grasping.

Developing grasping strategies that not only navigate the environment, but also exploit constraints to improve grasping success is considered a challenge, which applies to the deployment of modern robots in diverse pick-and-place scenarios [Newbury 23, Babin 21, Elguea-Aguinaco 23]. Previous devices have shown the capability to perform EC grasping motions via pre-programmed trajectories with both rigid objects [Xu 09, Odhner 12] and fabric [Koustoumpardis 14]. This chapter builds upon these previous approaches of EC grasping by using the observations of Eppner *et al.* [Eppner 15] to formulate a reward signal that encourages the gripper of Chapter 4 to execute *arbitrary grasping a flattened garment laying upon a rigid, level surface*. Such an approach has never been explored, and the research of this chapter demonstrates that a data-driven method is a viable approach to learn *arbitrary* grasping, bringing robots closer to generalised manipulation. As fabric conforms to applied manipulations, kinematic information and sensor feedback from the gripper will inform the RL algorithms about the current manipulation stage. Developing this skill with RL also provides a baseline approach to investigate more complex fabric handling tasks. For example, a difficult task investigated by Ono *et al.* [Ono 01] required a robotic system to perform a biomimetic grasp on a stack of fabric while only extracting the topmost item. An RL approach could potentially be applicable to such a challenge.

5.1.2 Scope for Reinforcement Learning

This chapter considers two state-of-the-art RL algorithms for training intelligent robotic behaviour. The algorithms of Soft-Actor Critic (SAC) [Haarnoja 18c] and Twin Delayed DDPG (TD3) [Fujimoto 18] have seen success in learning a desired policy assuming the appropriate world and reward function are present. A policy in RL refers to a module that uses the present state to generate an action. In this case, the policy assumes that an external vision component has detected a garment, and a motion planning algorithm has moved the gripper to pin the fabric to the environment surface as shown in Figure 5.2. The contribution of this chapter investigates how TD3 and SAC can learn *arbitrary* grasping and apply such a skill to the gripper of Chapter 4. Such research requires the construction of both an RL environment and reward function. The developed environment uses the URDF model from Section 4.3.5, alongside a simulated Baxter robot in PyBullet [Coumans 16]. This simulated learning environment follows the standard template from the Python Gym library [Brockman 16], enabling a smooth integration of RL algorithms for training. Approaching EC grasping in the scenario of Figure 5.2 uses several observations from anthropomorphic grasping discussed in Section 5.1.1. First, there is the aforementioned aspect of distance traversed across the environment. The state vector in this learning problem will include a value that holds the distance the ‘fingertip’ has travelled along the environment while dragging inwards to a grasp. In this case, the ‘fingertip’ refers to the position of the triaxial force sensor (see Equations 4.12-4.15). The distance notion also forms part of the reward signal discussed in Section 5.2.2, where the gripper will receive a reward for dragging along the environment before closing into a grasp.

Heinemann *et al.* [Heinemann 15] discussed how the ‘closing’ action, while exploiting the environment, usually takes on a similar form before conforming to static object characteristics. In the learning context of this chapter, the manipulated garment will conform to the gripper’s manipulation actions. Considering this aspect, the simulated learning environment does *not* include fabric while learning the EC grasping policy. Rather the policy will learn EC motions upon a rigid surface with varying friction properties. Simulating fabric remains complex and while some research has seen success in deploying fabric with PyBullet [Matas 18, Seita 21], the objective of this chapter is to create EC grasping policies that drag along an environment before closing into a grasp. Integrating deformable objects into a simulated RL environment lies beyond the intended scope of this project. The learning of this chapter focuses upon whether a reward signal based around the distance notion of human-centric EC grasping can result in RL policies that can grasp flattened garments via the learned motions.

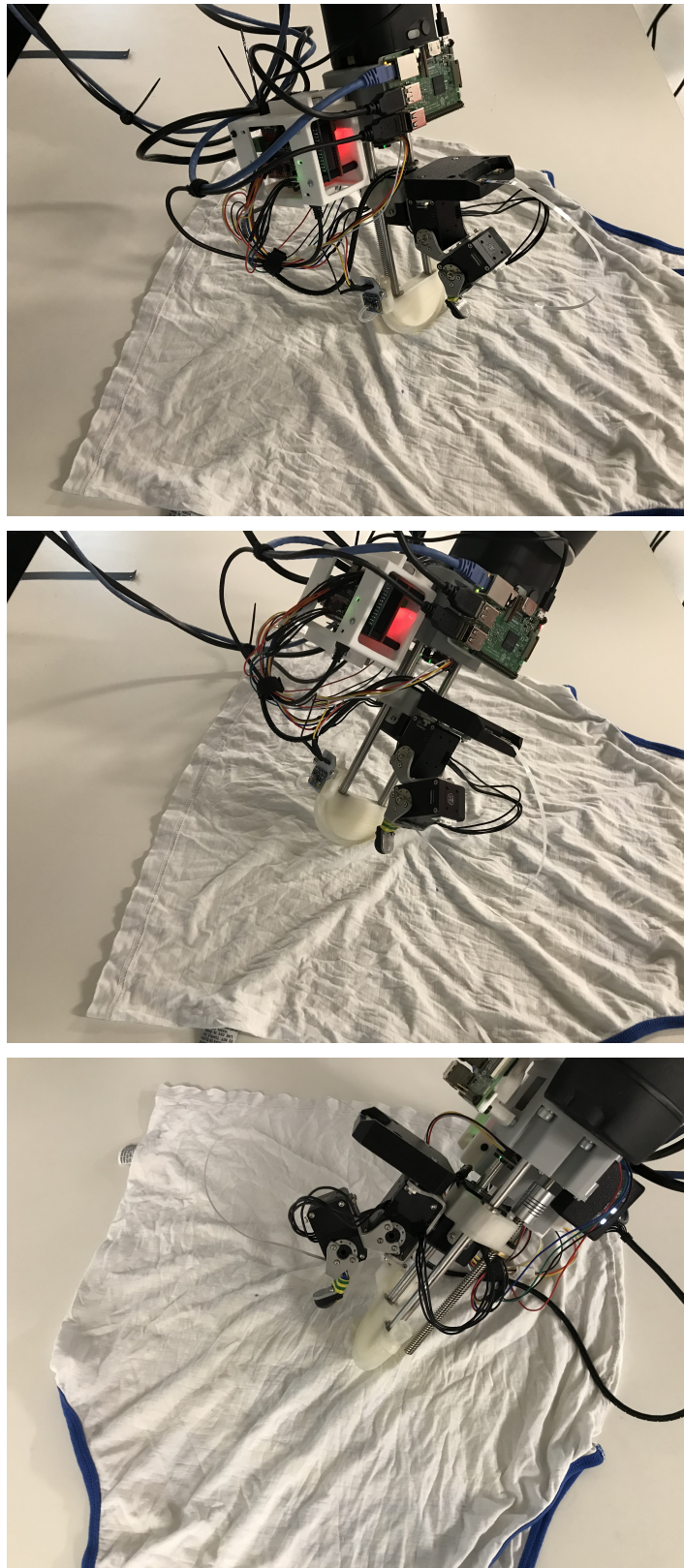


Figure 5.2: Example scenarios that the RL problem will address by learning *arbitrary* grasping, showing the gripper pinning fabric to the table surface from a range of wrist orientations.

After training in simulation, this chapter deploys learned policies to perform EC grasping with the hardware platform upon several garments. This chapter demonstrates that these learned motions in simulation translate to the grasping of fabric with the real-world gripper. However, there are complexities associated with EC grasping. As previous studies have shown, the friction interaction between the gripper, fabric, and environment will influence the forces required to execute a biomimetic grasping motion and affect grasp success. Koustoumpardis *et al.* [Koustoumpardis 14] and Shibata *et al.* [Shibata 09] both sought to change the friction factor between their respective grippers and fabric by overlaying silicon on their ‘fingertips’ while performing a pin-and-drag motion. To ensure that the training process encounters a range of friction interactions, the simulation performs domain randomisation on the friction properties of both the gripper’s triaxial force sensor and the environment surfaces. In addition, the simulation ensures that the policy is robust to perturbations in collision interactions by slightly varying the orientation of the rigid surface on which the policy learns EC grasping. Such features can assist in the training of robust policies that can grasp fabrics with differing physical properties and also aid learned skills in overcoming the reality gap.

5.2 Implementation

5.2.1 Learning Algorithms

This section presents the implementation of the RL algorithms TD3 [Fujimoto 18] and SAC [Haarnoja 18c]. Both have successfully trained specific robot skills within locomotion and manipulation, as discussed in Section 2.3.3. However, the implementation of RL algorithms can significantly impact the training process. The case study of on-policy algorithms by Engstrom *et al.* [Engstrom 19] demonstrated how code-level-optimisations impact results more significantly than the algorithm of choice. Furthermore, Henderson *et al.* [Henderson 18] studied RL training aspects including hyperparameter tuning, reward scaling, seeding configurations and NN structures, observing how they could significantly impact both on-policy and off-policy approaches. In an effort to remain transparent while discussing the training of EC grasping upon the gripper of Chapter 4, this section outlines the implementation details of both TD3 and SAC. For convenience during development, both algorithms follow a single learning process. Additionally, the algorithm development process made use of the PyTorch library for practical convenience and access to CUDA-compatible hardware. Upon initialisation, the learning process activates specific components related to each algorithm. However, updating the network parameters differs for

both approaches as they use distinct objective functions and contain some fundamental differences, as outlined further in this section. Before discussing these differences, this section provides an overview of the generalised learning framework.

General Off-Policy Implementation

Algorithm 1 presents the pseudocode of the learning procedure TD3 and SAC follow for learning EC grasping. The process uses several hyperparameters defined in notes under Algorithm 1 and comments in the pseudocode. The learning process follows standard conventions used in off-policy algorithm implementations. The process begins by executing trajectories that use random-action transitions which the replay buffer stores. After collecting enough of these random steps, the policy samples actions from the algorithm throughout executed trajectories. Simultaneously, the learning process updates policies after each trajectory by sampling transitions from the replay buffer. As the learning algorithm converges to a solution, the actions taken by the policy improve upon the total reward received throughout trajectories. Algorithm 1 also outlines elements from the environment described further in Section 5.2.2. For example, after R_s steps have occurred since the last *hard-reset*¹ event, another *hard-reset* occurs that resets certain simulation elements and performs domain randomisation on the robot manipulator and surrounding environment.

The learning algorithm \mathcal{A} contains functions including UPDATE and GETACTION, which are both components of SAC and TD3, with minor distinctions due to the algorithms' implementation differences. The set of learning hyperparameters \mathcal{S} outlined in Algorithm 1 differ depending on whether the learning process is using TD3 or SAC. In order to track and compare different algorithm progression, the learning procedure also periodically saves the NN weights using an in-built function of the PyTorch library. While training, the function EVALUATIONPROCEDURE uses Tensorboard to log learning progress. The learning framework calls EVALUATIONPROCEDURE every E_s steps to perform an evaluation sequence involving 16 grasping actions equally distributed across the range of possible wrist orientations in simulation. The evaluation sequence returns the average reward received across these grasp attempts and Tensorboard saves this data throughout training.

¹The *hard-reset* function is discussed in Section 5.2.2.

Algorithm 1 General Off-Policy EC learning Learning Process

```

function EXECUTELEARNINGTRAJECTORY( $\mathcal{A}, I_s, N_s, \mathcal{D}$ )
  for each environment step do
     $s \leftarrow \text{GETSTATE}()$ 
    if  $N_s \leq I_s$  then
       $a \leftarrow \text{SAMPLERANDOMACTION}()$   $\triangleright$  Use random actions until  $N_s$  exceeds  $I_s$ 
    else
       $a \leftarrow \mathcal{A} \rightarrow \text{GETACTION}(s)$ 
    end if
     $\text{SENDACTION}(a)$   $\triangleright$  Sends actuator commands to the robot manipulator
     $s' \leftarrow \text{GETSTATE}()$ 
     $r \leftarrow \text{REWARD}(s, a, s')$ 
     $\mathcal{D} \rightarrow \text{STORE}(s, a, s', r)$   $\triangleright$  Stores a complete transition to the replay buffer
     $N_s + = 1$ 
  end for
  return  $N_s, \mathcal{D}$ 
end function
Input: ( $T_s, I_s, \mathcal{D}_{len}, \mathcal{S}, rs$ )
   $N_s \leftarrow 0$   $\triangleright$  Sets the number of steps taken to 0
   $\mathcal{A} \rightarrow \text{INITIALISE}(\mathcal{S})$   $\triangleright$  Initialises the learning algorithm  $\mathcal{A}$  (Can be TD3 or SAC)
   $\mathcal{D} \leftarrow \text{REPLAYBUFFER}(\mathcal{D}_{len})$   $\triangleright$  Sets up the replay buffer  $\mathcal{D}$  of length  $\mathcal{D}_{len}$ 
   $\text{SETRewardSCALING}(rs)$   $\triangleright$  Setup the reward scaling for the environment
  while  $N_s \leq T_s$  do  $\triangleright$  Learning processes performs  $T_s$  environment steps
     $N_s, \mathcal{D} \leftarrow \text{EXECUTELEARNINGTRAJECTORY}(\mathcal{A}, I_s, N_s, \mathcal{D})$ 
    if  $N_s > I_s$  then
      for  $U$  number of iterations do
         $\mathcal{A} \rightarrow \text{UPDATE}(\mathcal{D})$   $\triangleright$  Training begins after all random steps are collected
      end for
    end if
    if  $R_s$  steps have occurred since the last hard-reset event then
       $\text{ENVIRONMENTHARDRESET}$   $\triangleright$  Perform a hard-reset event
    end if
    if  $E_s$  steps have occurred since the last evaluation procedure then
       $\text{EVALUATIONPROCEDURE}$   $\triangleright$  Perform an evaluation procedure
    end if
    if  $M_s$  steps have occurred since the model saving process then
       $\text{SAVEPOLICIES}$   $\triangleright$  Save the actor and critic NN weights
    end if
  end while
   $\text{SAVEPOLICIES}$   $\triangleright$  Save final actor and critic NN weights upon completion.

```

rs - Refers to the reward scaling hyperparameter.

T_s - Refers to the maximum number of steps the training process performs.

N_s - Refers to the number of steps executed.

I_s - Refers to the initial number of random steps collected during training.

R_s - Refers to the number of steps taken before a new *hard-reset* action is called.

E_s - Refers to the number of steps taken before a new evaluation procedure is called.

M_s - Refers to the number of steps taken since the models were last saved for Tensorboard analysis.

U - Refers to number of updates applied to the RL algorithm after a trajectory.

\mathcal{S} - Refers a set of hyperparameters related to the initialisation of \mathcal{A} . See details about each specific algorithm in Section 5.2.1.

TD3 Algorithm

TD3 by Fujimoto *et al.* [Fujimoto 18] is the first off-policy approach used to investigate the developed environment for learning EC grasping. A brief description of TD3 is presented in Section 2.3.2. Algorithm 2 presents the implementation details and mathematical definitions required while using TD3. These details also include definitions for pseudocode functions specific to TD3 from Algorithm 1, namely INITIALISE, GETACTION and UPDATE. TD3 uses six NN function approximator components, which converge to an optimal behaviour solution using the reward schema from Section 5.2.2. This set of neural networks consist of the critic networks ($Q_{\theta_1}, Q_{\theta_2}$), their target variants ($\bar{Q}_{\theta_1}, \bar{Q}_{\theta_2}$), the policy network (π_{ϕ}), and the target policy network ($\bar{\pi}_{\phi}$). The two critic networks are present as TD3 applies a technique known as clipped double Q-learning, which takes the minimum Q-value estimate between the two target critics while updating, thereby avoiding overestimation. The NN architectures follow simple structures defined in PyTorch. The critic networks follow a three-layered structure that takes in a vector $\{s + a\}$ and outputs a Q-Value as shown in Figure 5.3(b). The policy follows a similar structure using a state vector s to generate an action a . Figure 5.3 displays these network structures alongside the activation functions of each layer. The final output layer on the policy function NNs uses a *tanh* activation function, bounding the output between -1 and 1 , before re-scaling the output to the action and observation vector's normalised range between -0.75 and 0.75 (Section 5.2.2).

TD3 is an expansion to the DDPG algorithm [Lillicrap 15], which enables the learning of continuous control tasks in dynamic environments. Therefore, many of the aspects that update the various NN modules come from definitions present in DDPG alongside the related algorithms of Deep Q-Network (DQN) by Mnih *et al.* [Mnih 13] and Deterministic Policy Gradient (DPG) by Silver *et al.* [Silver 14]. Algorithm 2 displays the loss function for updating the critic networks ($Q_{\theta_{1,2}}$), which uses a form of the Mean-Squared Bellman error (MSBE) [Sutton 18]. The Bellman error, sometimes referred to as the Bellman residual, is the difference between the target Q-value estimate (denoted as y) and the Q-value $Q_{\theta_i}(s, a)$. The estimation of the target Q-value y uses an augmented action value \tilde{a} , which derives from the target policy $\bar{\pi}_{\phi}(s')$ with noise applied from a normal distribution. Fujimoto *et al.* [Fujimoto 18] suggest applying the additional noise to sampled target policy actions as a method which smooths the critic network updating process. Finally, the last step of clipped double-Q learning parses the augmented action value \tilde{a} through both target critic networks, after which the minimum value is chosen to calculate y .

The policy takes in a state vector and generates an action vector using the NN structure visualised in

Algorithm 2 TD3 Learning Functions

```

function INITIALISE( $\mathcal{S} = \{B_s, \gamma, E_n, N_c, P_n, P_{del}\}$ )
  Initialise policy ( $\pi_\phi$ ) and critic ( $Q_{\theta_1}, Q_{\theta_2}$ ) neural networks with weights  $\phi, \theta_1$  and  $\theta_2$ .
   $\bar{\pi}_\phi \leftarrow \pi_\phi, \bar{Q}_{\theta_1} \leftarrow Q_{\theta_1}, \bar{Q}_{\theta_2} \leftarrow Q_{\theta_2}$   $\triangleright$  Assign target variant weights from base neural networks
  Save hyperparameter set  $\mathcal{S}$  to local object memory
  return
end function
function GETACTION( $s$ )
   $a \leftarrow \pi_\phi(s) + \epsilon, \epsilon \sim \mathcal{N}(0, c \times E_n)$ 
  return  $a$ 
end function
function UPDATE( $\mathcal{D}, N_s$ )
   $N \leftarrow \mathcal{D} \rightarrow \text{GETBATCH}(B_s)$   $\triangleright$  Obtain  $B_s$  transitions  $(s, a, s', r)$  from the replay buffer
   $\tilde{a} \leftarrow \bar{\pi}_\phi(s') + \text{CLIP}(\epsilon, -c, c), \epsilon \sim \text{CLIP}(\mathcal{N}(0, P_n), -N_c, N_c)$   $\triangleright$  Augmented action  $\tilde{a}$  created
   $y \leftarrow r + \gamma \min_{i=1,2} \bar{Q}_{\theta_i}(s', \tilde{a})$   $\triangleright$  Estimate target value  $y$  using minimum target critic estimate
  Update critics  $Q_{\theta_i} \leftarrow \text{argmin}_{Q_{\theta_i}} N^{-1} \sum (y - Q_{\theta_i}(s, a))^2$   $\triangleright$  Update critics via MSBE loss
  if  $N_s \bmod P_{del} == 0$  then
     $\nabla J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$   $\triangleright$  The policy loss function using DPG
    Update the policy  $\pi_\phi(s) \leftarrow \nabla J(\phi)$ 
    Update the target networks via ‘soft updates’
     $\bar{Q}_{\theta_i} \leftarrow \tau Q_{\theta_i} + (1 - \tau) \bar{Q}_{\theta_i}, \text{ for } i = \{1, 2\}$ 
     $\bar{\pi}_\phi \leftarrow \tau \pi_\phi + (1 - \tau) \bar{\pi}_\phi$ 
  end if
  return
end function

```

Hyperparameters

B_s - Batch size: The number of transitions sampled by the replay buffer while updating.

γ - Discount factor: A constant value representing the impact of future rewards upon learning.

E_n - Exploration noise: A float defining a normal distribution when sampling actions while training.

N_c - Noise clipping: A constant that limits the maximum amount of noise while updating the policy.

P_n - Policy noise: A float defining a normal distribution for noise applied to actions from $\bar{\pi}_\phi(s')$.

P_{del} - Policy delay: An integer constant used to delay the policy and target network updates.

Other components

$\mathcal{N}(m, \mu)$ - Generates values from a normal distribution with a mean of m and standard deviation of μ .

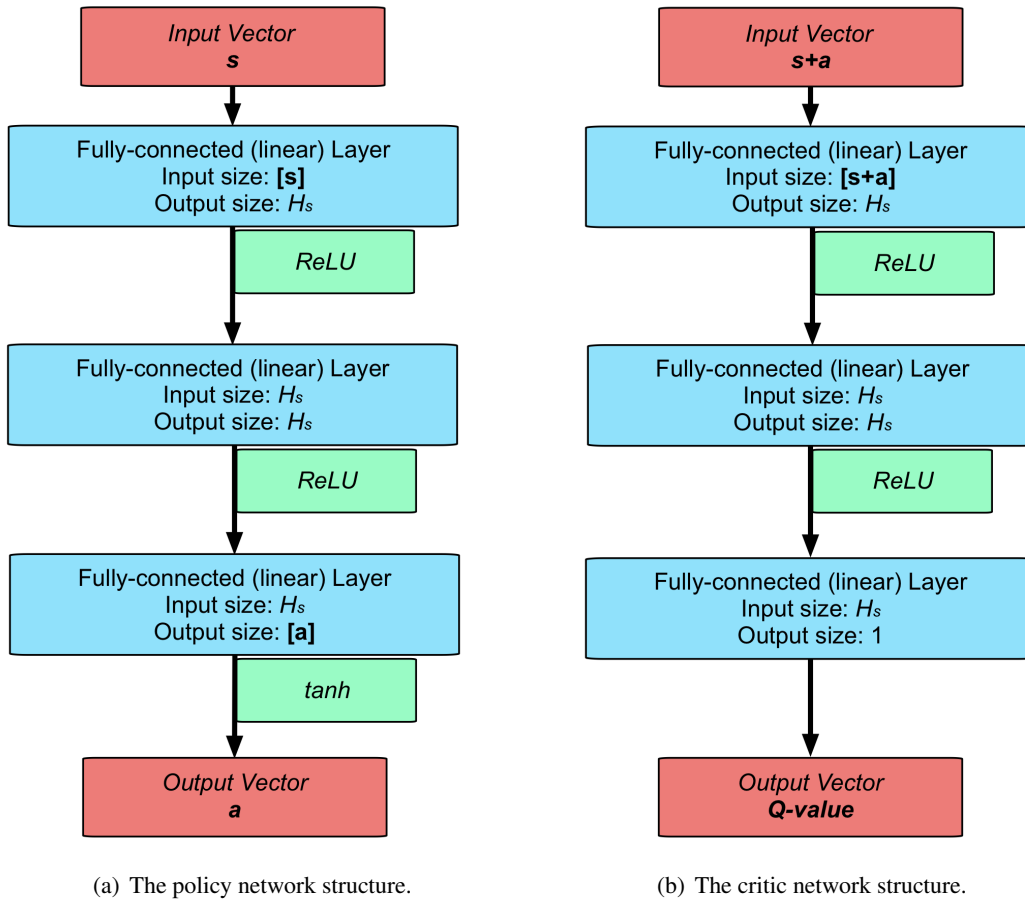
$\text{CLIP}(i, l, u)$ - Limits an input value (i) to a range ($l-u$).

c - A value that defines the maximum action available from the environment.

\tilde{a} - Refers to actions subjected to additional noise.

MSBE - Mean-Squared Bellman error

5.3(a) and uses a loss function based on the DPG algorithm [Silver 14]. A ‘soft update’ approach updates the weights of the target networks by tracking their learned counterparts using the hyperparameter τ as shown in Algorithm 2. Another suggestion by Fujimoto *et al.* [Fujimoto 18] was to delay the update of the policy and target function approximators using P_{del} to reduce overestimation bias and stabilise training. For the implementation of the RL algorithms in this chapter, both the critic (Q_{θ_i}) and policy

Figure 5.3: Network Structures used in TD3.²

(π_ϕ) networks use the ADAM Optimiser [Kingma 14] when updating their weights.

SAC Algorithm

In addition to TD3, this chapter also implements SAC to evaluate performance and demonstrate that the built environment of Section 5.2.2 is compatible with both off-policy approaches. Haarnoja *et al.* [Haarnoja 18b] presented SAC around the same time TD3 became available, before publishing an improved version [Haarnoja 18c], this updated version is implemented in this chapter. Similar to TD3, SAC is an off-policy actor-critic algorithm. However, the RL objective of the actor and critic components use the maximum entropy framework (MEF), which further impacts the derivation of the loss functions for the NNs. In addition, SAC uses a stochastic policy rather than the deterministic approach of TD3. One can describe the SAC as an algorithm, which learns a task while acting as randomly as possible. Like TD3, SAC follows the learning process outlined in Algorithm 1. Algorithm 3 presents the unique

²The variable H_s refers to Hidden Size, a hyperparameter which determines the input/output size of the internal neural network layers, defined in Table 5.1.

components of SAC integrated into the learning process.

SAC follows a similar structure to TD3 when comparing NN modules. There are two critic function approximators ($Q_{\theta_{1,2}}$) alongside a actor network (π_{ϕ}). There are also target variants of the critic NNs. However, unlike TD3 there is no target policy network. While developing the SAC implementation for EC grasping, the critic networks followed the same structure from TD3, Figure 5.3(b). However, while hyperparameter tuning and attempting to optimise SAC’s performance, this chapter found adding another layer to the policy NN slightly improved learning. Therefore, the policy network for SAC takes on a slightly different structure which Figure 5.4 visualises. The update of the target variants also uses ‘soft updates’. Incorporating the MEF into an actor-critic algorithm introduces some fundamental changes. If one considers a standard RL objective as the maximisation of the expected

³The variable H_s refers to Hidden Size, a hyperparameter which determines the input/output size of the internal neural network layers, defined in Table 5.2.

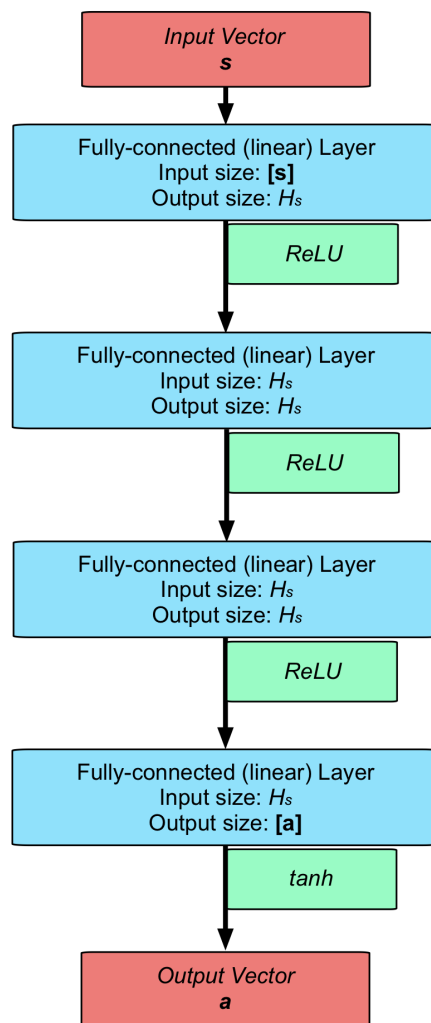


Figure 5.4: Network Structures used in TD3.³

Algorithm 3 SAC Learning Functions

```

function INITIALISE( $\mathcal{S} = \{B_s, \gamma, \bar{\mathcal{H}}, T_{del}\}$ )
  Initialise policy ( $\pi_\phi$ ) and critic ( $Q_{\theta_1}, Q_{\theta_2}$ ) neural networks with weights  $\phi, \theta_1$  and  $\theta_2$ .
   $\bar{Q}_{\theta_1} \leftarrow Q_{\theta_1}, \bar{Q}_{\theta_2} \leftarrow Q_{\theta_2}$   $\triangleright$  Assign critic target variant weights from base neural networks
  Save hyperparameter set  $\mathcal{S}$  to local object memory
  return
end function
function GETACTION( $s, Eval=False$ )
  if Eval then
    return  $a \leftarrow \pi_\phi(s)$ 
  else
    return  $\bar{a} \sim \pi_\phi(s)$ 
  end if
end function
function UPDATE( $\mathcal{D}, N_s$ )
   $N \leftarrow \mathcal{D} \rightarrow \text{GETBATCH}(B_s)$   $\triangleright$  Obtain  $B_s$  transitions  $(s, a, s', r)$  from the replay buffer
   $\bar{a}' \sim \pi_\phi(s')$ 
   $y \leftarrow r + \gamma(\min_{i=1,2} \bar{Q}_{\theta_i}(s', \bar{a}') - \alpha \log(\pi_\phi(\bar{a}'|s')))$ 
  Update critics  $Q_{\theta_i} \leftarrow N^{-1} \sum (y - Q_{\theta_i}(s, a))^2$   $\triangleright$  Update critics via MSBE loss
   $\bar{a} \sim \pi_\phi(s)$ 
   $\pi_\phi \leftarrow N^{-1} \sum (\alpha \log(\pi_\phi(\bar{a}|s)) - \min_{i=1,2} Q_{\theta_i}(s, \bar{a}))$   $\triangleright$  Update policy
   $\alpha \leftarrow N^{-1} \sum (-\alpha \log(\pi_\phi(\bar{a}|s)) - \alpha \bar{\mathcal{H}})$   $\triangleright$  Tune Entropy temperature
  if  $N_s \bmod T_{del} == 0$  then
    Update the target networks via ‘soft updates’
     $\bar{Q}_{\theta_i} \leftarrow \tau Q_{\theta_i} + (1 - \tau) \bar{Q}_{\theta_i}, \text{ for } i = \{1, 2\}$ 
  end if
  return
end function

```

Hyperparameters

B_s - Batch size: The number of transitions sampled by the replay buffer while updating.

γ - Discount factor: A constant value representing the impact of future rewards upon learning.

$\bar{\mathcal{H}}$ - The minimum entropy constraint: A value defining the constrained optimisation problem that tunes α

T_{del} - Target delay: An integer constant used to delay the target network updates.

α - Temperature: A hyperparameter dictating the impact of the entropy term upon learning.

Other components

\bar{a} - Refers to actions generated from the policy with the reparameterisation trick.

Eval - A boolean that indicates whether the generated action will use the reparameterisation trick, *False* if generating actions that are stored in the replay buffer or created when updating the policy.

MSBE - Mean-Squared Bellman Error

rewards, $\sum_{t=1}^T \mathbb{E}_{(s_t, a_t) \sim \pi_\phi} [r(s_t, a_t)]$, SAC introduces the additional term of $\alpha \mathcal{H}(\pi_\phi(\cdot|s_t))$ to the objective, where \mathcal{H} represents the entropy measure and the temperature parameter α determines the impact of the entropy measure on the learning objective. Such changes result in the new learning objective

$$J(\pi) = \sum_{t=1}^T \mathbb{E}_{(s_t, a_t) \sim \pi_\phi} [r(s_t, a_t) + \alpha \mathcal{H}(\pi_\phi(\cdot|s_t))].$$

SAC’s integration of the MEF into results in a modified loss function that updates the soft Q-function update, called the soft Q update. As Haarnoja *et al.* [Haarnoja 18c] discuss, the subsequent version of SAC uses two critic networks to mitigate positive bias in the policy improvement step, similar to TD3 [Fujimoto 18]. The loss function of the critic components follows a similar method to the critic of TD3. However, the entropy term is considered when calculating the target Q-value y . As shown in Algorithm 3, the term $-\alpha \log(\pi_\phi(\bar{a}'|s'))$ appends to the minimum estimate from the target Q-functions. Haarnoja *et al.* [Haarnoja 18c] refer to this expression as the soft Bellman residual. Algorithm 3 reflects this change that minimises the soft Bellman residual with a loss function that uses the Mean-Squared Error. The articles of Haarnoja *et al.* [Haarnoja 18b, Haarnoja 18c] provide the appropriate proofs and details for deriving the soft Bellman residual from the entropy-enhanced learning objective.

The update of the policy uses a modified loss function that also considers the MEF. The policy improvement step guides the policy towards greedy behaviour with respect to the exponential of the soft Q-function by minimising the Kullback-Leibler (KL) divergence, such an approach results in the policy loss function $\alpha \log(\pi_\phi(\bar{a}|s)) - \min_{i=1,2} Q_{\theta_i}(s, \bar{a})$ [Haarnoja 18b, Haarnoja 18c]. The policy optimisation also makes use of the reparameterisation trick to ensure sampled actions are differentiable with respect to the policy weights ϕ . Similar to TD3, all NN updates make use of the ADAM optimisation algorithm [Kingma 14].

Finally, the learning process of the initial SAC version by Haarnoja *et al.* [Haarnoja 18b] could be brittle to the temperature hyperparameter α , which dictates the entropy impact as the algorithm learns. The subsequent version of SAC by Haarnoja *et al.* [Haarnoja 18c] presents automated entropy adjustment approach that tunes α throughout training. This approach formulates a constrained optimisation problem that satisfies a minimum entropy constraint ($\bar{\mathcal{H}}$), while maximising the expected return. In practice, an ADAM optimiser object updates α with the loss function displayed in Algorithm 3.

5.2.2 Formulation

The development of this thesis presents custom versions of TD3 and SAC within ROS packages to execute the training process for EC grasping on the gripper. Both algorithms are off-policy approaches that learn by collecting transition steps into a replay buffer, which the learning process samples throughout training. These algorithms have seen success in learning skills such as turning a valve [Ahn 20], locomotion [Zhu 22, Ahn 20] or mobile manipulation [Wang 20a]. However, these RL algorithms require both an environment and reward function to discover the optimal policies.

Building a Learning Environment

PyBullet [Coumans 16] hosts the deep learning environment for EC grasping and uses a ROS interface to communicate with the RL algorithms. URDF files imported into PyBullet construct the simulation's physical components. Objects in the simulation include a Baxter robot with the lateral grasp-inspired gripper attached to each hand and an environment surface to execute EC grasping motions upon. The simulation incorporates several domain randomisation aspects, changing various simulation properties that can potentially assist trained policies deployed in the real world by making them robust to dynamic variance. Aspects randomised include simulation gravity, actuator velocities, communication latency, the mass of the gripper links, and the friction properties of interacting components. Appendix D.1.2 presents details surrounding the randomised components and their range of variance.

The environment surface uses four different URDF models with differing surface patterns subjected to friction domain randomisation to provide a range of grasping surfaces with diverse contact interactions. Figure 5.5 visualises the learning environment with these varying environment surfaces and the robot agent. From a coding implementation perspective, an RL environment requires the functions of *step* and *reset* and handles the creation of state and action spaces. The *step* function parses an action a to create a transition tuple consisting of the previous state, last action taken, present state and received reward (s, a, s', r) . The environment's response to taken actions creates the state variables and informs the received reward r . The *reset* function resets the simulator environment for a new learning trajectory. In addition, the environment includes a modified *hard-reset* function which resets the simulation and imports a new URDF environment surface while performing domain randomisation upon elements of the simulation (Appendix D.1.2).

The RL formulation of learning EC grasping assumes an episodic nature, with the beginning of a trajectory defined when the gripper pins the static plate into the environment surface, and the end occurs once the manipulator reaches a grasping configuration after dragging along the environment. A grasping configuration occurs when the sensor is registering force values and it's position is upon the static plate (calculable with FK expressions). Each trajectory while learning has a timeout limit of 25 seconds, which will cause the trajectory to terminate if exceeded, this timeout is reduced to 7 seconds when executing evaluation trajectories that gauge learning progress.

The environment class structure defines the action and state vectors upon initialisation. The action space is the eight commands required to control the gripper, and the observation vector consists of 24 elements. The observation vector includes present joint positions, velocities, the forward kinematics (to the

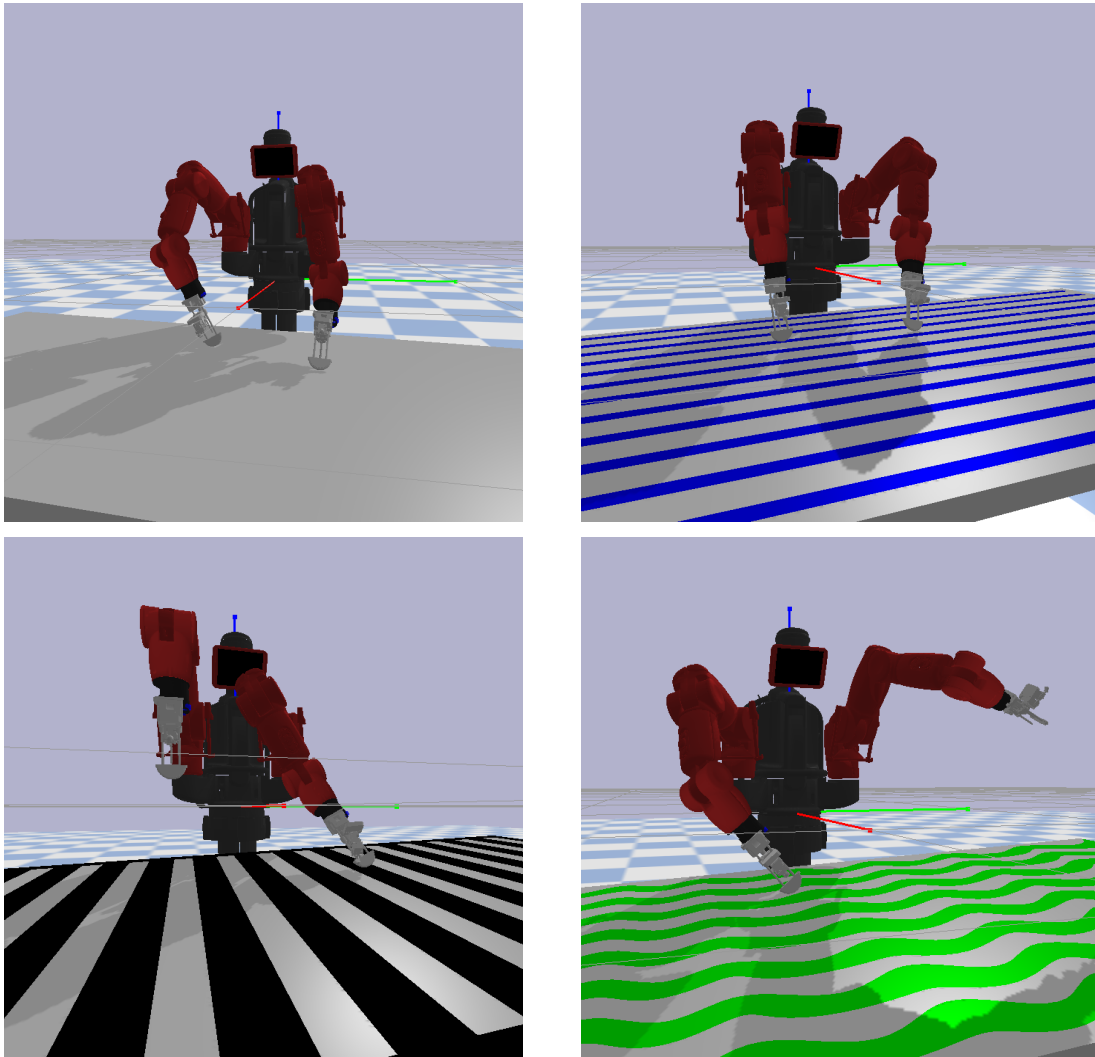


Figure 5.5: The learning environment in PyBullet with various surface textures.

sensor position), the present commands targets (i.e. target joint position, speed and current), the force sensor readings, wrist orientations (roll-pitch-yaw angles, *RPY*) and the distance travelled while navigating the environment. Like the observation space vectors defined in the ROBEL benchmark suite [Ahn 20], the state vector includes the last actions taken. The logic behind this choice was due to the partial observability common in robotic systems (Section 2.3.3). In addition, the commands sent to the actuators will influence manipulator behaviour and play a role in the present state of the device. These observation and action vectors normalise their values between -0.75 and 0.75 before being processed by the NN function approximators. Appendix D.1.1 presents the state and action variable information alongside the value range used for the normalisation process.

The operation of the environment during learning uses a ROS framework to transmit the latest feedback from the gripper through a custom message format, see Figure 4.16. Such an approach simplifies

the integration of policies onto hardware as the ROS interface remains consistent regardless of platform. This approach also enables the simulation node to use in-built protections that restart the program if failures occur during long term training. In addition, ROS action servers and service requests communicate data between the learning process and environment behaviour. For example, the *reset* function of the environment calls a ROS action server, which informs the PyBullet simulation to reset the pose of the robot arm that last performed a learning trajectory. The policy also communicates with the agent (simulated gripper) at a rate of 10Hz .

Throughout learning, the environment will place both arms of the simulated Baxter robot into a position where a learning trajectory can occur. While training, a single trajectory will occur with the gripper upon one of the arms. Upon resetting, the gripper on the alternative Baxter arm will perform another trajectory, while the original arm is resetting its pose. This process continues throughout training, with both arms placing the gripper in a random pose for a new learning trajectory. If performing a *hard reset*, an action server call informs the simulation node to remove and re-import all URDF files with a domain randomisation step. Figure 5.6 visualises the ROS framework for training and controlling the simulation.

This chapter establishes policy convergence to an *arbitrary* grasping solution, a learning challenge that requires the orientation of the manipulator. The observation vector s uses *RPY* values representing this orientation. For the simulated environment and hardware deployment, it was convenient to represent the wrist orientation through the Baxter robot's URDF file and in-built kinematic solvers. In addition, this approach used an established coordinate system provided by the Baxter's ROS packages and URDF descriptors that provided uniformity across the simulation and hardware. The *RPY* values were taken from the transform sequence between the Baxter robot's base frame and wrist pose of each arm. In addition, these *RPY* values derive from the rotation matrix using the Cardanian *ZYX* sequence [Corke 17]. This method was convenient as the only relevant information for learning *arbitrary* grasping was the manipulator's orientation relative to the environment surface, which lies parallel to the xy plane of the Baxter's base frame. Therefore, the observation vector discards the first rotation value from the Cardanian *ZYX RPY* calculation and only uses the remaining y and x rotation values. Thereby only using the rotation data relevant to the manipulator's orientation relative to the environment surface.

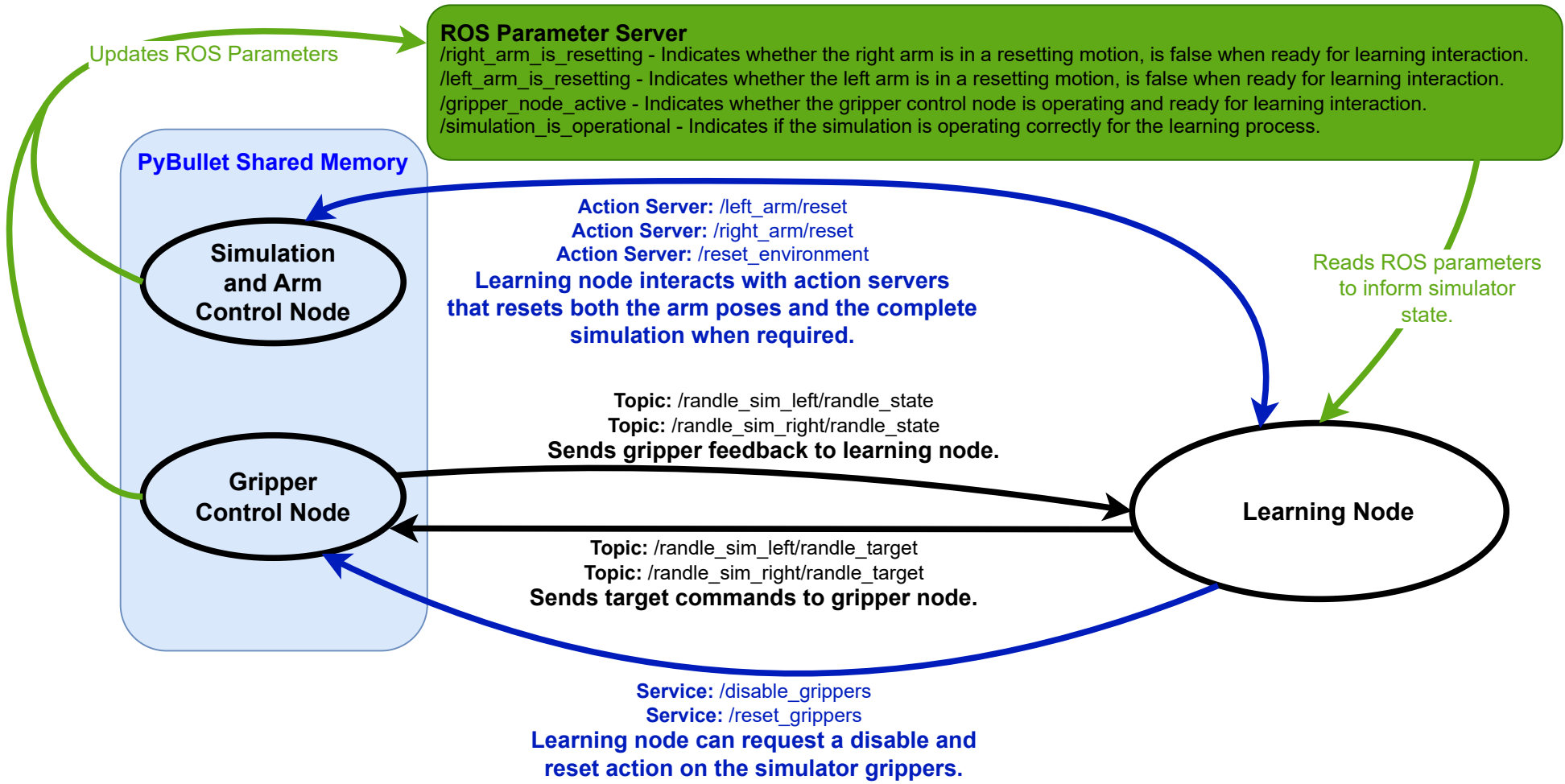


Figure 5.6: The ROS framework that interfaces the simulation with the off-policy learning process.

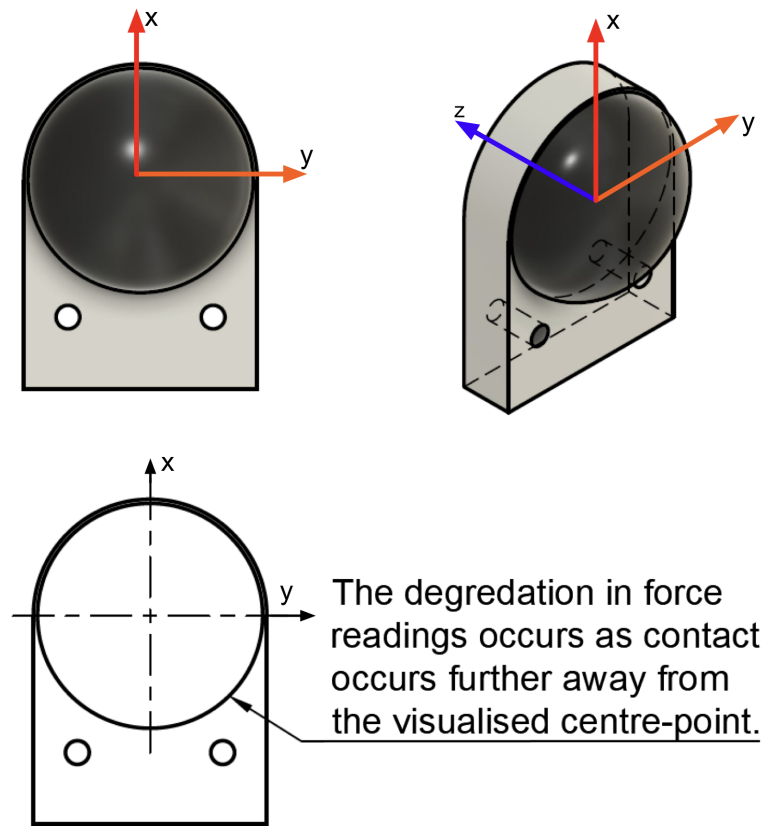


Figure 5.7: The triaxial force sensor with its sensing coordinate frame and centre-point highlighted.

Developing the simulation was an iterative process that encountered the *reality-gap* (introduced in Section 2.3.3) challenges one expects when training policies in simulation deployed directly onto hardware. Domain randomisation techniques can assist with these challenges. However, one particular hurdle presented difficulties when transitioning from the simulation to the real world. The sensor attached to the TCP of the manipulator uses deformations in quasi-deformable materials to estimate exerted forces, resulting in several behaviours that were difficult to replicate in simulation. Figure 5.7 displays the sensor, its measurement coordinate frame and a visualised centre-point of the rubber component. As contact moved away from this centre-point, non-linear behaviours in the force readings became apparent, which could not be replicated in simulation. However, PyBullet does allow developers to access collision locations and exerted forces within rigid bodies, which allows users to write custom data pre-processing functions. Following hardware experimentation and an assessment of simulation capabilities, the optimal solution to move the simulator closer to the hardware's behaviour emerged: solely utilising the force readings from the z -axis of the sensor (see Figure 5.7).

While this approach closed the gap between the sensor's real world behaviour and the simulation,

such a methodology still required further considerations. The visualised centre-point of Figure 5.7 is where the sensor’s force readings are most accurate, as this location is where the device’s manufacturing calibration sequence took place. However, as contact points move away from the centre, the force read in the z -axis degrades and decreases compared to the actual force applied. Examination of the sensor discovered that if contact upon the rubber occurred further than $4mm$ from this centre-point, the sensor would not register applied forces upon the z -axis. As PyBullet can simulate rigid bodies force interactions and collision locations, the simulated learning environment replicates these behaviours. To develop robust policies, the domain randomisation function changes an exponential expression defining the degradation behaviour of the simulated sensor readings. Thereby exposing the policy to a wide range of non-linear force readings throughout training.

Formulating a Reward Function

The reward function provides a mechanism that guides the RL algorithms to execute *arbitrary* grasping by providing a numerical reward or penalty between states. The applied RL algorithms attempt to execute a behaviour that maximises the received reward across a trajectory. Using the observations outlined in Section 5.1.1, which discusses observations from Eppner *et al.* [Eppner 15] and previous grippers, the formulated reward function encourages the gripper’s ‘fingertip’ to make contact with the environment surface, traverse across the surface and close the gripper into a grasp pose. If following this desired sequence of motions, the learning algorithm will receive several positive reward values. Alternatively, if the gripper performs a behaviour moving away from a grasping configuration or abandons a traversing motion along the environment surface, the reward function will return negative penalising values.

The reward function is a class structure that tracks information throughout the trajectory, including the accumulated reward while grasping, whether contact with the environment is occurring, and the distance traversed across the environment. This stored information then informs the received reward for each transition step. A *reward-shaping* methodology assists the learning algorithms in converging to the desired EC grasping motion. The gripper begins each trajectory with the bumper sensor of the static plate in contact with the environment and the gripper in a default configuration as visualised in Figure 5.2. These rewards are a function of the triaxial force sensor and the gripper’s forward kinematics that consider the position of the sensor (t) relative to w given by Equations 4.12, 4.13, 4.14 and 4.15⁴. Figure 5.8 displays the essential coordinate frames used by the reward function with h defining the position of

⁴ t and A_s from Chapter 4 are the same coordinate frame.

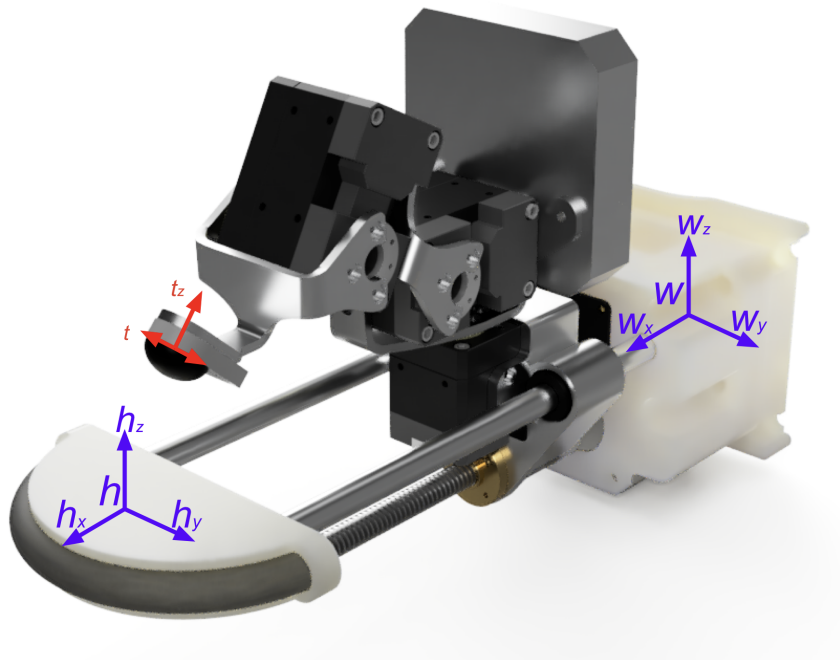


Figure 5.8: A visualisation of key components of the gripper learning setup.

the gripper's static plate.

Upon beginning a trajectory sequence, the only reward the gripper can receive is when it makes initial contact with the environment. Defining this action simply uses the position of t and readings from the triaxial force sensor. If the sensor registers a force value along t_z above or equal to $0.8N$ and the sensor's position is significantly above the location of the static plate, the reward considers this interaction as contact with the environment. Developing the reward function was an iterative process and found that a threshold of $0.8N$ was acceptable. To determine whether the sensor's position was above the plate, the z displacement of t relative to w (Equation 4.14) had to exceed $3mm$.

Once the gripper has made contact with the environment, rewards or penalties occur based on traversing actions. Traversing actions are transitions between states where contact with the environment is still present, meaning that the sensor is still reading a force value along t_z above $0.8N$, while fulfilling the kinematic conditions outlined in the previous paragraph. However, the position of t relative to h has changed. If the sensor has moved closer to h , the reward function returns a positive value based on the Euclidean distance reduced between t and h . Alternatively, if this Euclidean distance between t and h has grown, the reward function returns a penalising value reflecting this increase.

The class structure hosting the reward function tracks the Euclidean distance between t and h traversed by the manipulator throughout a surface dragging motion. If the gripper abandons a traversing

motion and lifts the sensor from the environment surface, the reward structure resets this distance to 0. Before closing into a grasp, a traversing motion must reduce the Euclidean distance between t and h by $15mm$ from the initial contact point while dragging along the environment. Assuming that the gripper has met the inward distance traversing requirements, the final component of the grasping trajectory is the grasp closure motion. Grasp closure is detected when the sensor's position is upon the static plate and the sensor registers an applied force greater than $1N$.

There are a number of circumstances, which could interrupt a grasping attempt. Examples include closing into a grasp without traversing inward along the environment for the required $15mm$ or abandoning a traversing action by lifting the sensor from the environment surface. Under these situations, the reward function penalises the positive accumulated reward from the traversing action. In addition, the distance traversed and other grasping details gathered throughout the grasping motion are reset to 0 such that the reward structure is setup for a new grasping attempt within the episode. The reward function returns 0 when the gripper is in a grasping configuration without interacting with the environment and when the gripper is searching for initial contact with the environment. Algorithm 4 in Appendix D.2 provides pseudocode, which outlines the reward function alongside the specific values it returns under various circumstances.

5.3 Results - Simulation

Both TD3 and SAC performed training in the simulated environment of Section 5.2.2 across five seeds, Figure 5.11(a). Figure 5.9 visualises the training progression for each seed under TD3. TD3 learned to perform EC grasping under the hyperparameters outlined in Table 5.1 by colliding with the environment and traversing inwards to a grasp across the five seeds displayed in Figure 5.9. However, converging to this solution was a time-intensive process. As Figure 5.9 displays, converging to the grasping solution could take between 25 and 200 hours. With further hyperparameter tuning, the algorithm may converge in a more time-efficient manner. These figures also show the learning process converged to a solution that usually received a reward of approximately three in evaluation procedures, given the reward scaling (rs) value of 0.015 used.

Unfortunately, SAC did not see the same success TD3 achieved when attempting to learn EC grasping. Figure 5.10 visualises the training process. Despite an exhaustive parameter search, the training runs conducted in this thesis were unable to converge SAC to appropriately learn EC grasping reliably across seeding, Figure 5.11(b). Figure 5.10 shows how two of the seeds (**1547** and **2307**) managed to converge to partial-solutions that were improving their evaluation scores. However, these training runs were still unable to converge to a complete grasping solution, resulting in policies that saw an evaluation score of roughly 1.5. Had a successful grasping solution been found, this score would have exceeded a value of 2, and usually expected to float between 3 and 4. The remaining seeds failed to converge to a solution, with evaluation rewards ranging between 0-0.5. Upon closer inspection, several of these policies converged to solutions that avoided negative penalties, resulting in policies that took no actions and simply did not interact with the environment. An example, which demonstrates such a solution best, is the training run with a seed value of **9008**.

TD3 being capable of converging to an EC solution while SAC struggles could be due to several factors. First, SAC is an algorithm that usually places a strong emphasis on exploration by incorporating the entropy term into the objective function. While beneficial in certain environments, the constructed environment of Section 5.2.2 may not have required a high degree of exploration. While training, the automated tuning of the hyperparameter α^5 usually converged to a near-zero value across all SAC training runs within the first 200,000 steps, further reinforcing this notion. The value of 0.95 given in Table 5.2 refers to the initial value of α before the tuning process throughout training occurred.

In addition, the deterministic policy of TD3 and its exploration methods may have been more suited

⁵This hyperparameter is a value that determines the impact of the entropy term upon training.

to the simulated EC grasping environment. As previously mentioned throughout this chapter, biomimetic grasping motions take on a similar form [Heinemann 15], and the possibility remains that learning the motions of *arbitrary* grasping, even with the different trajectories required for the various wrist orientations, may have been an RL problem where the entropy incorporation of SAC hindered the learning process. There are further hyperparameter tuning options that could improve the performance of SAC in the simulated learning environment. For example, this implementation of SAC sets the minimum entropy constraint ($\bar{\mathcal{H}}$) hyperparameter to the value of $-\dim(a)$, the negative dimension of the action space, as previous authors have recommended [Haarnoja 18c, Wang 20b]. Exploring alternatives for $\bar{\mathcal{H}}$ alongside adjusting parameters including the reward scaling rs , neural network architectures and other hyperparameters may yield an improved training process for SAC in the simulated environment of Section 5.2.2.

The training processes in simulation also highlighted a number of limitations in the EC grasping environment that could benefit from improvement in future developments. To start with, domain randomisation for the table surface friction, force sensor body friction, and the table orientation all use a pseudo-random number generator (PRNG) algorithm to generate environment variables upon each environment *hard-reset*. While such an approach was acceptable to learning EC grasping, especially for TD3, a hand-crafted approach to ensure that the environment traversed across a wide range of dynamic variables may improve the learning process. Figure 5.9 shows that in some training runs, the evaluation process sees a significant drop in score. Indicating that the learning process occasionally encountered environmental conditions in which the policy struggled to perform. By ensuring that a comprehensive range of dynamic environment conditions were encountered, this training process may improve the policy’s resilience to environmental factors. Despite these limitations, the learned policies under TD3 remain acceptable to evaluate on the hardware platform, grasping fabric using learned policies directly from the simulated environment.

Table 5.1: The hyperparameters of applied TD3 algorithm

Hyperparameter	Value
B_s The batch size, a hyperparameter that dictates how many samples are taken from the replay buffer during NN updates.	256
E_n Exploration noise, a constant that defines the standard deviation of a normal distribution which samples noise appended to policy-derived actions.	0.05
γ The discount factor, a constant value that determines the impact of future rewards beyond the present transition.	0.99
R_s Number of steps taken before a new environment <i>hard-reset</i> event occurs.	4000
H_s Hidden size, a hyperparameter that dictates the input and output sizes of NN layers that the entry or output points of the function approximator.	256
lr Learning rate, a constant value that governs the pace of NN optimisation.	0.0001
N_c Noise clip, a value that limits noise applied to augmented actions that TD3 uses while training the critic networks.	0.25
T_s The total number of environment transition steps performed throughout the learning process.	5000000
P_{del} Policy delay, a constant that delays policy and target network updates for a certain number of steps.	2
P_n A float constant defining the standard deviation of a normal distribution..	0.1
D_{len} Maximum size of the replay buffer.	1000000
rs Reward scaling, a constant multiplier applied to received rewards from the environment.	0.015
τ A constant defining the rate of the ‘slow-moving update’ process for the target networks.	0.005
U Update steps performed after a trajectory.	250
I_s The number of initial steps in which the policy samples random actions.	1024
M_s The number of episodes between saving NN weights.	600
E_s The number of steps between evaluation procedures.	12000

Table 5.2: The hyperparameters of applied SAC algorithm

Hyperparameter	Value
B_s The batch size, **See definition in Table 5.1**	1024
γ The discount factor, **See definition in Table 5.1**	0.97
α The initial temperature parameter value, determines the impact of the entropy term upon the learning objective.	0.95
R_s **See definition in Table 5.1**	5000
H_s Hidden size, **See definition in Table 5.1**	256
\bar{H} The minimal expected entropy, a value defining the minimum entropy constraint.	$-dim(a)$
lr Learning rate, **See definition in Table 5.1**	0.0003
lr_α Learning rate for the ADAM optimiser that tunes the entropy term α	0.00005
T_s **See definition in Table 5.1**	10000000
T_{del} Target delay, An integer constant used to delay the target network updates	2
D_{len} **See definition in Table 5.1**	1000000
rs Reward scaling, **See definition in Table 5.1**	0.02
τ **See definition in Table 5.1**	0.005
U **See definition in Table 5.1**	250
I_s **See definition in Table 5.1**	4096
M_s **See definition in Table 5.1**	1000
E_s **See definition in Table 5.1**	50000

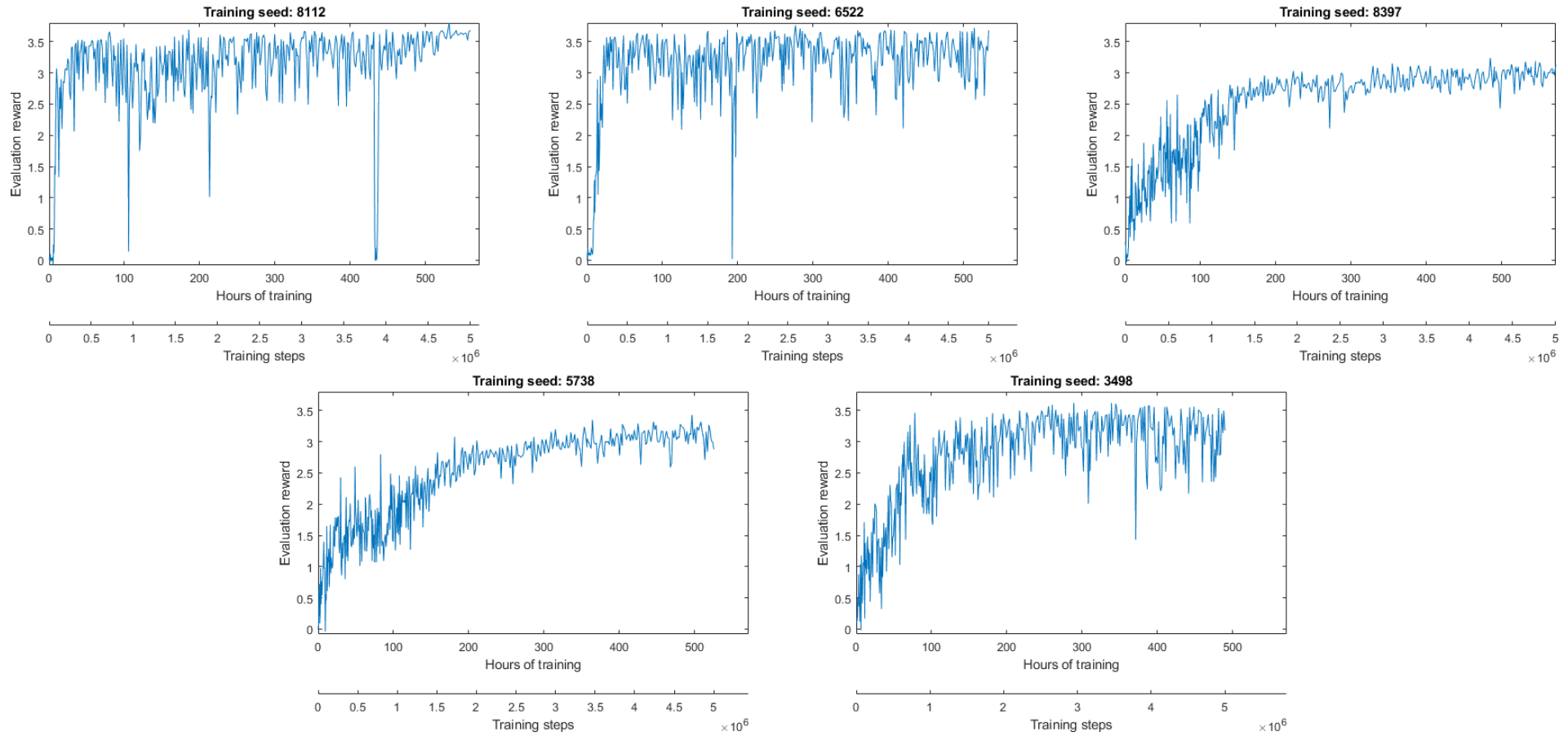


Figure 5.9: The hours of training and steps taken while learning with TD3 under various seeds using hyperparameters from Table 5.1.

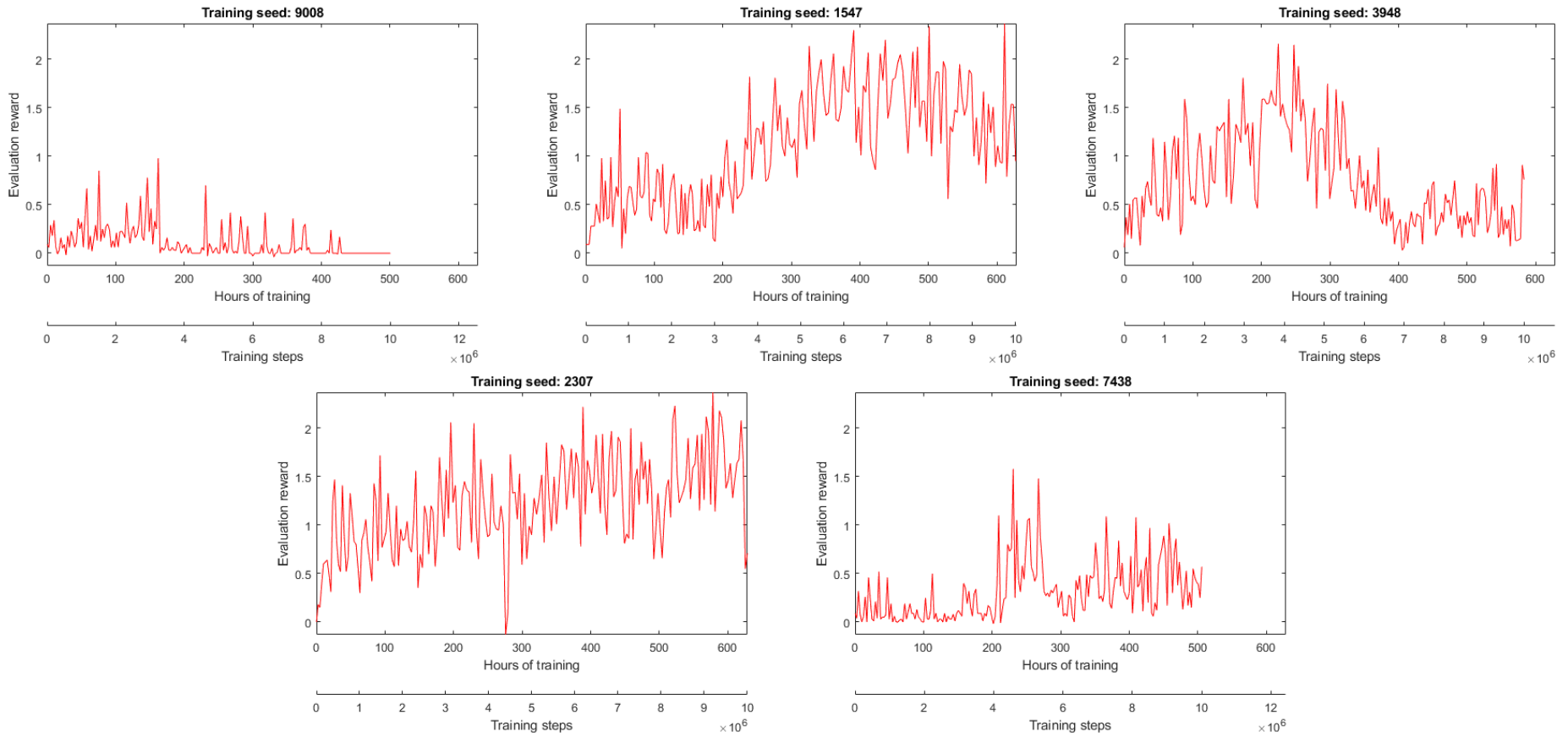
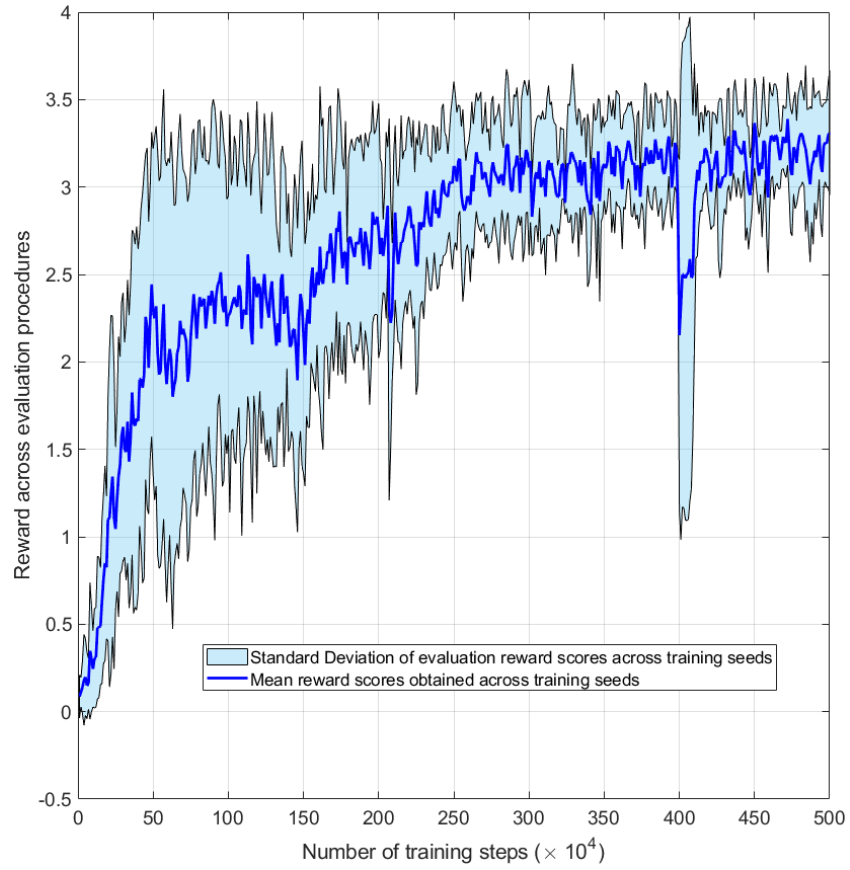
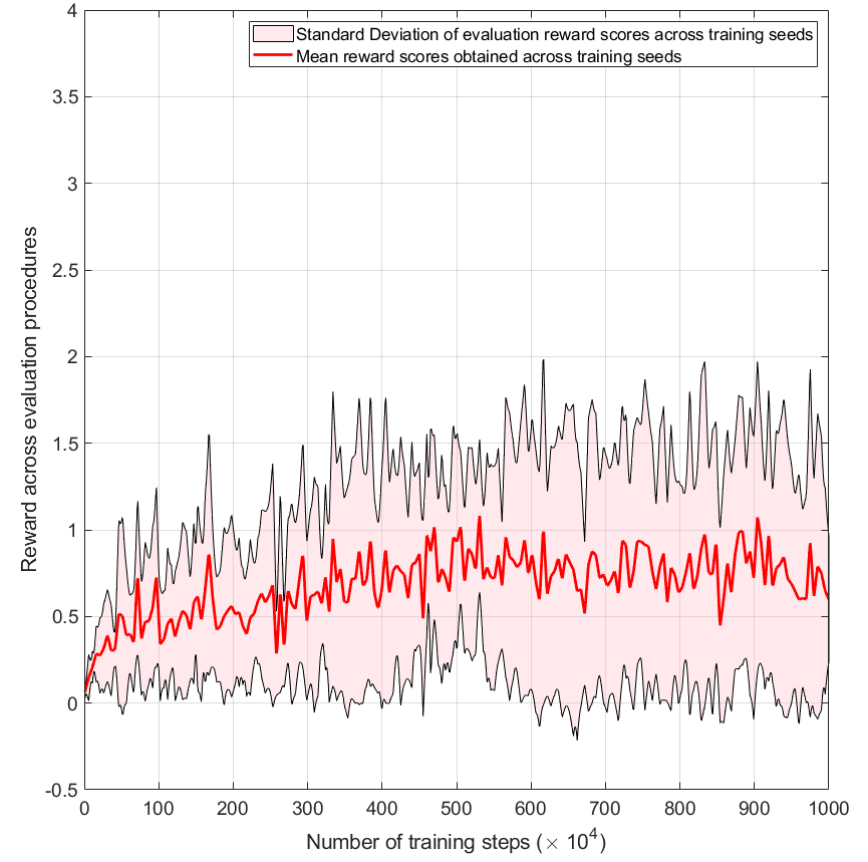


Figure 5.10: The hours of training and steps taken while learning with SAC under various seeds using hyperparameters from Table 5.2.



(a) TD3



(b) SAC

Figure 5.11: The average reward, along with the standard deviation visualised, is shown across the training seeds for each algorithm.

5.4 Results - Hardware Deployment

A key aspect of this chapter is the deployment of policies from the simulated URDF model to the hardware platform. As the implementation of SAC struggled to converge to an EC grasping solution, this chapter directly deploys TD3 (seed **5738**) to the hardware and evaluates EC grasping in the real world. Such a method encounters challenges associated with the *reality-gap*, a degradation in performance when transferring simulation-trained policies into the real world. The hardware evaluation took place by placing a flattened garment on a table beneath the gripper⁶. The Baxter robot would move the gripper to pin the fabric to the table before letting the trained policy control the gripper, executing a grasp. Initial grasping motions were executed upon the hardware to ensure that the evaluation could take place safely on the hardware before this evaluation took place. This process resulted in several heuristic adjustments were made to the hardware prototype and the sensor feedback data. First, the gripper would sometimes place the triaxial sensor in a position where all environmental contact occurred at the edge where the rubber met the rigid body of the sensor. As the rubber and rigid body connect via an industrial adhesive, a small 3D-printed cover was placed over this region to prevent incidental contact prying the rubber off the rest of the sensor.

While grasping from hardware, it remained simpler to extrapolate the wrist orientation component from transforms defined by the Baxter robot. As the wrist orientation forms part of the state vector (see Section 5.2.2) and the simulated environment used the Baxter's URDF transform data, this approach was a convenient method to transfer the learning process into the real world. In addition, the policies deployed on hardware use augmented triaxial force sensor readings to improve EC grasping motions. In the simulated environment of Section 5.2.2, force readings upon the z-axis of the sensor (see in Figure 5.7) greater than $0.8N$ were considered an environment interaction transition assuming kinematic conditions were met. The sensor degrades in accuracy as contact moves away from the centre-point visualised in Figure 5.7, therefore the hardware deployment applied Equation 5.1, in which F_x , F_y and F_z are the original sensor readings of the triaxial force sensor. F_{z_a} parses through the neural network during training. Figure 5.12 visualises the value of m_t given F_z lies between 0.4 and 1.44 when applying Equation 5.1. The policy trained in simulation recognises read F_z values greater than $0.8N$ as interactions traversing the environment, enabling the policy to treat lower real world values of F_z as similar traversing events for improved EC grasping.

⁶Associated media files of the hardware deployment can be found at the author's personal website: <https://robodave94.github.io/>.

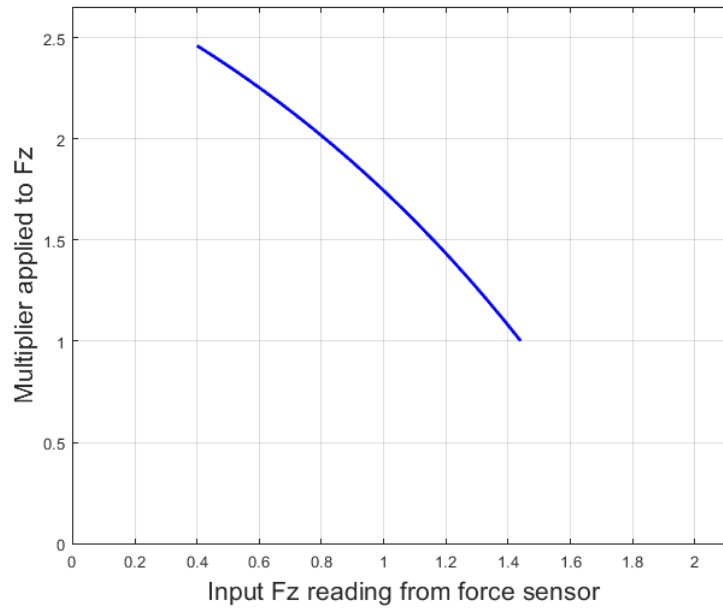


Figure 5.12: A plot of the exponential multiplier (m_t) applied to Fz from the real world sensor data given Fz lies between 0.4 and 1.44.

$$m_t = \begin{cases} (-1.1(1.4^{2Fz}) + 3.9) & \text{if } 0.4 \leq Fz \leq 1.44 \\ 1 & \text{otherwise} \end{cases} \quad (5.1)$$

$$Fz_a = m_t \cdot Fz + \frac{F_x}{2} + \frac{F_y}{2}$$

The grasping evaluation occurred across 16 wrist orientation positions with six different garments, see Figure 5.13. Each grasp was attempted three times, therefore, this hardware evaluation attempts 288 EC grasping trajectories on the hardware. Figure 5.14 visualises one of these EC grasping attempts upon a scarf. Table 5.3 displays the results of EC grasping with the fabricated gripper. The amount of transition steps taken is reflected in each cell representing a grasp attempt. Similarly to the simulation environment, the policy communicates with the hardware platform at a rate of $10Hz$. The Table highlights the rate of successful grasps alongside displaying the duration of the grasping attempts across wrist orientations and garments. In addition, Table 5.3 highlights the wrist orientation of the manipulator while grasping with the variables Wz and Wy . These variables represent radian values applied to the wrist orientation. Thereby moving the gripper towards the flattened garment in a diverse range of positions. If one considers frame w from the DH parameters of Table 4.1, the orientation of the manipulator while grasping, is calculable via the transform sequence $w \cdot R_y(\frac{\pi}{2})R_z(Wz)R_y(Wy)$. At a high level, this sequence represents orienting the gripper downwards towards the table before applying the rotations of Wz and Wy sequentially.



(a) Long-Sleeved Black Top



(b) Scarf



(c) Heavy Jumper



(d) Jean Shorts



(e) White-Collared Shirt



(f) Black Chino Trousers

Figure 5.13: The target garments for EC grasping with the real-world gripper.

Table 5.3: Results of hardware EC grasping using TD3.

Angle code	Wz (rad)	Wy (rad)	Black Cotton Shirt			Red Scarf			Thick Jumper			Jean Shorts			White-Collared Shirt			Black Chino Trousers			Angle Success Rate (%)	Angle Success Duration (s)
			T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3		
A	-0.552	-0.524	58	94	63	63	52	94	71	48	32	54	74	44	42	42	55	58	48	43	0.94	5.9
B	-0.184	-0.524	181	63	73	34	115	59	47	86	33	42	68	46	33	41	88	60	57	43	0.89	6.39
C	0.184	-0.524	41	50	33	36	63	64	115	65	44	45	38	45	37	97	36	93	71	35	1	5.6
D	0.552	-0.524	46	47	159	169	82	63	87	85	70	26	87	41	120	117	80	81	40	94	0.94	8.29
E	-0.552	-0.367	104	67	49	71	29	66	71	39	77	53	48	39	46	59	38	41	64	43	1	5.58
F	-0.184	-0.367	37	40	31	24	44	56	48	34	70	56	59	38	48	39	40	35	31	50	1	4.33
G	0.184	-0.367	33	35	73	37	79	33	122	71	69	63	40	33	31	32	51	37	102	29	0.83	5.17
H	0.552	-0.367	40	50	41	63	250	65	39	48	37	48	39	62	49	40	37	148	52	37	0.89	5.29
I	-0.552	-0.209	75	28	197	53	58	29	95	77	98	43	59	83	174	50	51	32	42	221	0.94	8.16
J	-0.184	-0.209	34	70	37	38	32	65	43	67	118	62	77	137	112	106	27	39	33	59	0.94	6.41
K	0.184	-0.209	69	48	33	35	57	28	61	28	59	25	56	26	31	75	28	81	92	40	0.78	5.12
L	0.552	-0.209	68	71	144	250	143	73	184	73	74	91	66	64	89	70	27	49	62	129	0.78	8.97
M	-0.552	-0.052	31	62	43	28	109	27	53	52	18	27	48	103	101	76	60	30	29	36	0.94	5.33
N	-0.184	-0.052	250	76	35	126	40	64	33	36	26	55	250	63	55	29	46	42	107	23	0.89	5.35
O	0.184	-0.052	37	96	33	69	36	206	115	87	212	26	87	74	32	25	24	36	24	42	0.83	6.05
P	0.552	-0.052	156	126	250	250	250	69	65	250	81	75	250	250	250	116	250	67	250	250	0.44	9.44
Trial Success Rate (%)			0.94	0.94	0.94	0.88	0.88	1	0.88	0.69	0.88	0.69	0.69	0.75	0.94	1	0.94	0.94	0.94	0.94		
Garment Success Rate			0.94			0.92			0.82			0.71			0.96			0.94				
Garment Success Average Duration (s)			6.73			6.47			6.33			5.98			5.87			5.88				

Notes

The integer taken for each grasp indicates the number of steps taken. Transition steps occurred at 10Hz. Every grasp attempt was given a 25 second time limit.

Grasps highlighted in red indicate cases where the gripper did not grasp the material correctly.

Each garment grasping process underwent three trials and each trial is referenced with the bold text on the second row. For example, **T1** refers to Trial 1.

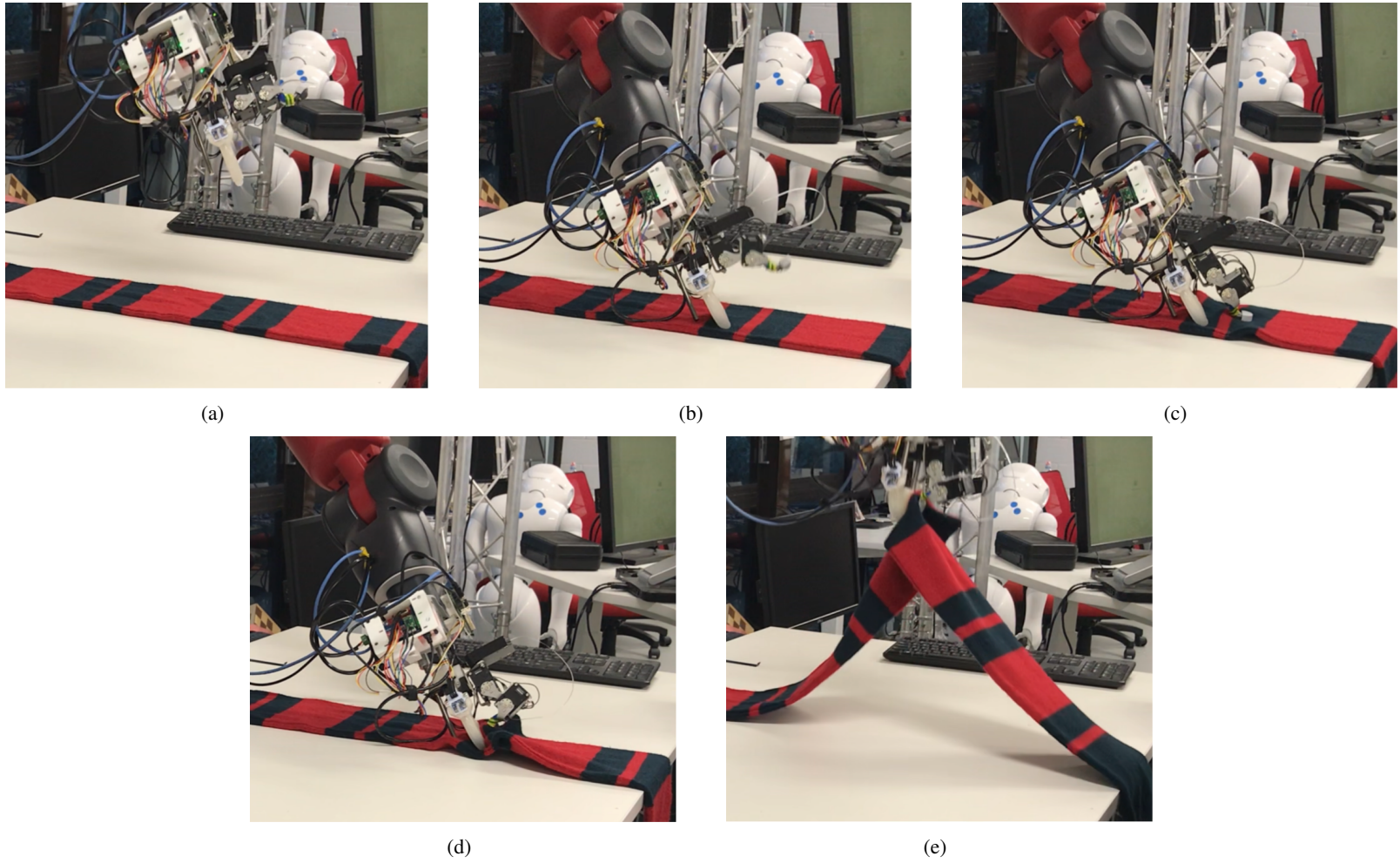


Figure 5.14: The gripper grasping a scarf using a TD3 policy trained only in the simulation.

As Table 5.3 shows, the policy when deployed on hardware grasped flattened clothing from the environment at a success rate above 70%. Interestingly, the average successful grasping duration across garments varied within a second, with the white-collared shirt exhibiting the lowest value of 5.87 seconds and the black cotton shirt taking 6.73 seconds. However, when comparing the duration of successful grasps across wrist orientations, the average duration varied within a range of around 5 seconds, with angle F taking 4.33s and angle P taking 9.44s. These combined observations reinforce that EC grasping motions use similar forms and that the evaluated policy had learned to grasp more effectively for specific wrist orientations. The actual target garments had little impact on the duration of grasp. Furthermore, when comparing success rates across angles to the duration taken, there does not appear to be a strong correlation between the time taken and the success rate. However, the angle with the lowest successful grasping rate (P) took the most prolonged average duration to grasp.

While the garments did not demonstrate variability when considering the duration taken, the success rates across clothing items ranged between approximately 71% success for the jean shorts and 96% for the white-collared shirt. Several observations presented themselves surrounding the proposed RL approach while deploying TD3 upon the hardware. Failure could occur for a number of reasons. In some cases, the torque of the gripper's thumb appendage was capable of lifting the Baxter's robot arm, which would lift the static plate from pinning the garment to the table surface. Such a situation would result in the gripper sliding fabric along the environment surface rather than producing protrusions. Alternatively, certain collision interactions would cause the stepper motor to skip steps, affecting the kinematic calculations required for policy operation.

Some garments exhibited a low elastic energy and would hold the garment's state from applied manipulations. If the gripper attempted multiple biomimetic grasping motions, this property would improve grasping success as protrusions made by previous dragging motions would retain their shape and provide more regions on the garment's body for the thumb appendage to close the grasp. However, garments like the jean shorts demonstrated mechanical properties with a higher elastic energy due to the type of material, the weave or the garment's construction properties, which means that when protrusions or deformations made by previous dragging motions within a trajectory occurred, the garment would not hold this form and partially revert to the flattened state. This non-linear mechanical behaviour results in the gripper still completing grasping motions but producing less prominent protrusions to secure a grip on the garment. Table 5.3 shows how the jean shorts saw more failed grasp attempts while still completing the grasping motion, as failures occurred while performing under 250 transition steps.

5.5 Discussion

This chapter presented the inspiration, formulation, development and integration of a reinforcement learning approach towards environmentally constrained grasping. Driven by observed human-centric behaviour of dexterous grasping, a reward schema encouraging the gripper to traverse the environment resulted in simulation-trained policies able to grasp clothing in the real world with an accuracy above 70%. Research from the literature discussing grasping with environmental constraints [Heinemann 15, Eppner 15] inspired the reward schema and notion of EC grasping motions remaining similar in form, inferring that a simulation that learns the motions should remain applicable to grasping flattened clothing in the real world.

The total success rate was approximately 88% across the 288 hardware grasping motions performed. Despite the policies remaining capable of grasping in the real world, the average duration of successfully grasping flattened fabric was approximately 6.3 seconds and usually involved multiple dragging motions. In a practical deployment setting, this behaviour is not optimal and further work is required to refine the trained policies. However, considering the range of difficulties associated with the *reality-gap*, it is still encouraging to see the policies trained purely in simulation grasp fabric at such a high success rate when deployed to hardware.

5.5.1 The Resultant Model

This chapter focused on reinforcement learning and evaluation of off-policy models for EC grasping of clothing in the real world. Although the results were promising, there are limitations that must be addressed to make this approach practical. The training process for learning EC grasping in the simulated environment was time-intensive, as shown in Figures 5.9 and 5.10, with an average training time of 500 hours or more. However, Figure 5.11(a) shows that TD3 achieved a solution (on average) at around 350 hours before continuing to train for the user-determined number of steps (refer to T_s in Table 5.1). Furthermore, SAC's failure to find a solution highlights that off-policy approaches are not guaranteed to learn arbitrary grasping with the gripper of Chapter 4.

It is possible that the challenges faced during the learning process are not due to deficiencies in the process itself, but rather to the limitations of the sensor integration on the robot manipulator. As discussed in Section 5.2.2 and illustrated in Figure 5.7, it was difficult to replicate the behaviour of the TCP sensor in simulation. To address this, some compromises were made, resulting in the sensor having a small cone

of operation (approximately 10mm of available surface area contact on the sensor). As a result, rewards could only be reliably obtained when dragging inwards and when the small detection zone was in contact with the environment. Despite these limitations, the success of TD3 is encouraging, especially since no design optimisation process has been conducted on the gripper and sensor configuration.

After successfully deploying a trained TD3 policy to hardware, certain interesting aspects of the system came to light. As shown in Table 5.3, angle P took the longest to execute the grasp and showed the lowest success rate. Although a more detailed ablation study is needed, several possible reasons can be considered. One reason could be that creating random wrist orientations during training creates two linearly spaced vectors across the possible orientations of W_y and W_z . The wrist orientation is obtained by sampling these vectors without replacement. This approach may cause difficulty learning certain grasping aspects when the wrist orientation is close to the possible orientation limits. Another possibility is that the particular seed (5738) used on hardware did not effectively learn grasping at angle P .

5.5.2 Options for Further Exploration

After conducting this investigation, the resultant model presents several opportunities for further exploration that can lead to improved autonomous manipulation of fabrics. Unfortunately, the Ph.D. timeline restricted the research focus, so a broader scope was not possible right now. However, one immediate step to build upon this research is to continue learning the hardware in an analogous manner to Haarnoja *et al.* [Haarnoja 18a]. Alternatively, there are various training algorithms available to explore for learning EC grasping, including on-policy approaches [Schulman 17], demonstration-based learning [Rajeswaran 17], or image-based control [Yarats 21]. These approaches could be more effective in learning dexterous grasps that leverage the environment.

The first step before further improving the learning process is to upgrade the gripper’s TCP sensing configuration. The current sensing configuration on the fingertip is limited, which makes it challenging to create a hand-crafted reward signal. While the desired grasping behaviour was successful, an updated sensing configuration may benefit future expansions of this chapter. Due to the timeline and budget constraints within the Ph.D. program, integrating the triaxial force sensor as it currently stands was the appropriate course of action.

However, some considerations occurred during the development process which should be highlighted. The sensor could only detect traversing events when the centre point of the sensor or close regions registered contact. A fingertip-sensing configuration that detects applied forces across the en-

tire thumb's body could improve EC grasping. Additionally, a multimodal sensing device may assist in performing biomimetic grasping motions under more complex settings, such as extracting a single item from a stack or registering surface texture features. Expanding the design thinking framework to encompass sensing requirements could yield valuable insights surrounding EC grasping and broader textile manipulation.

5.5.3 Further Takeaways

The training in simulation and deployment to hardware highlighted several considerations for future investigations of EC grasping motions. RL is a challenge in balancing the exploitation vs. exploration issue within a learning process [Sutton 18]. From the simulated training results, it appears that learning EC grasping in the simulation environment of Chapter 5 may be a learning challenge requiring little exploration, as SAC, an algorithm known for emphasising exploration, failed to converge to a grasping solution. However, some adjustments could be made to the simulated environment to improve training, including handcrafted approaches to ensure a wide range of dynamic conditions were encountered, as opposed to the PRNG approach. In addition, the trained policies for TD3 and SAC could benefit from further investigation into simulation refinement, so it more closely resembles the hardware platform and fabric interactions in the real world. Alternatively, refining the hyperparameters further could yield results that encourage SAC to converge successfully in the environment of 5.2.2. Finally, the issue of excessive training run-times under the current setup remains present. Further hyperparameter tuning, code-level optimisations, or multi-agent implementations may assist in optimising the training duration.

The deployment to hardware also revealed a number of limitations, which one should consider when further exploring EC grasping behaviour on real-world systems. As mentioned in Section 5.4, the force exerted by the thumb appendage while pressing into the environment was sufficient to lift the arm, removing the static plate from the table and 'unpinning' the garment. In this particular case, the combination of Baxter robot's underlying control system combined with the gripper behaviour resulted in this situation occurring. However, it still highlights that the manipulator's actions will impact arm behaviour while performing biomimetic grasping in this case. Meaning that manipulator actions cannot be completely decoupled from the arm.

In addition, the open-loop nature of the prismatic rail driven by the stepper motor impacted real world grasping. Replacing the stepper motor with an appropriate closed-loop actuator would also improve hardware grasping success. Finally, while exploring EC grasping, niche scenarios saw the gripper failed

to grasp or produce a protrusion in the fabrics body due to the configuration of the garment near the grasp. Heavy regions of garments such as the waistband of pants impacted grasp success. To address this issue, exploration of computer vision algorithms to find the optimal grasp points on a garment's flattened body could present an interesting research challenge. Alternatively, EC grasping motions exerting a greater force against the environment is another viable approach.

5.6 Conclusion

The broader implications of this research indicate that EC grasping of a single flattened garment is a challenge solvable via data-driven approaches. The resultant RL policies and their capabilities were made possible by building a reward schema using human-centric characteristics observed in the literature [Eppner 15]. While limitations were present in the solution of this chapter, policies transferred directly to the real world proved capable of executing reliable grasping of various garments with differing mechanical properties. This success adheres to the observation of Heinemann *et al.* [Heinemann 15], who describe how the closing motions of EC grasping take on a similar form. The hardware evaluation saw variance in successful grasp rates between garments that ranged between 71% and 96%. Demonstrating that the diverse non-linear mechanical behaviour of different garments will have impact on grasping success. However, design refinement on the existing gripper and RL algorithm optimisation could see improvement building upon these initial results.

Expanding this reinforcement learning approach to EC grasping presents an opportunity to explore investigation avenues which could take many forms. An immediate example building upon the present learning configuration is the development of policies that learn *arbitrary* grasping with a TCP wrench control mechanism. The environment formulated in Section 5.2.2 uses a threshold value of $0.8N$. Assuming the correct kinematic conditions are met, registering a collision above this threshold is considered contact with environment. Adapting the RL algorithm to target varying force thresholds could result in more robust grasping policies. An example feature to explore this aspect includes *Hindsight Experience Replay (HER)* [Andrychowicz 17]. Further expansion research could investigate an alternative sensing setup as discussed in Section 5.5. Alternatively, Ono *et al.* [Ono 01] explore the complexities of biomimetic grasping motions that extract a single fabric item on a stack. Such a challenge could be addressed by building upon the approaches presented while incorporating new aspects into the RL formulation and sensing modalities.

This chapter concludes the research of this thesis. Building upon the efforts of Chapters 3 and 4, a human-inspired RL approach encouraged the gripper to perform grasps that leverage the environment with RL, which Eppner *et al.* [Eppner 15] describe as a fundamental capability contributing to the generalised manipulation capabilities of humans. There are a range of further human-inspired textile manipulation skills that require exploration, which could build upon the research conducted throughout this thesis.

Chapter 6

Thesis Conclusion

6.1 Research Summary

The motivating force behind this research was the current state of post-consumer textile waste. As a result of fast fashion and current industry practices, the ever-increasing amount of discarded clothing waste is damaging the environment, wasting the resources used to manufacture them, and contributing to CO_2 emissions through decomposition. Despite available chemical processes, which can recycle clothing, humans still send an enormous amount of garments to landfills and poorer nations as Section 1.1 outlined. The economic impracticalities of sorting and preparing clothing for recycling or resale with human labour remain a core contributing factor to this status quo. Therefore, this thesis suggests the development of robotic technologies that could perform laborious steps involved in textile preparation for recycling. In particular, the act of sorting garments by the type of object, colour or composition could automate the repetitive components within the manual labour required by recyclers. Chapter 2 presented a literature review investigating state-of-the-art robotic fabric manipulation. Throughout the literature, a consistent thinking pattern that describes humans as the most adept manipulators of fabric became apparent. The literature also found that current robot gripper designs for fabric manipulation were limited in functionality and design. These notions resulted in the research conducted throughout chapters 3, 4 and 5, which used anthropomorphic grasping characteristics to inspire, develop and train a distinctive robot manipulator targeting fabric manipulation. The design thinking framework was essential in translating human-inspired aspects into a robotic system, providing a stable workflow in which traditional engineering principles were applied to design, construct and train the gripper. After presenting the core research of this thesis, this final chapter now presents a review of research questions, limitations and

future considerations.

6.1.1 Addressing the Research Questions

Babin and Gosselin [Babin 21] presented a review on grasping and specifically discuss the multidisciplinary nature of robotic gripper development. On top of involving a series of engineering challenges surrounding mechanics, software development, electronics and design, one must also consider the intended applications and task scope. Gripper development is still a prominent area of research as no grasping solution has reached the generalised manipulation capabilities of human beings. A common approach is to develop specialised manipulators with limited skills for specific applications. Such an approach can balance the engineering challenges and applicability. This thesis has followed such methods to develop a gripper for textile pick-and-place sorting. Chapters 3, 4 and 5 each presented a core research question as listed below. The research involved in formulating, creating and training the gripper was a result of addressing these questions.

1. Chapter 3: *Can discussing previous grippers from a hand-centric, anthropomorphic viewpoint reveal unique limitations and highlight novel design inspirations?*
2. Chapter 4: *Can one use robotic development techniques to build and integrate a unique gripper based structurally around the lateral grasp?*
3. Chapter 5: *Can reinforcement learning algorithms execute environmentally constrained grasping by learning from a reward schema inspired by human-centric behaviour?*

Fabric manipulation is known to be complex [Sanchez 18]. Nevertheless, everyday humans fold clothing, fasten buttons, do laundry, place shirts on hangers, and dress themselves. Creating a generalised robotic agent that could perform all of these tasks would be a considerable undertaking beyond the current state-of-the-art. Many authors have recognised that humans are currently the most adept fabric manipulators [Koustoumpardis 04, Le 13]. While replicating the capabilities of humans when manipulating fabric is currently infeasible, many projects have sought aspects of human-inspired manipulation to incorporate into gripper design [Le 13, Donaire 20, Koustoumpardis 14] or robotic fabric handling strategies [Verleysen 20]. Some authors note that previous robotic gripper designs for fabric handling were limited in functionality or applicability [Donaire 20, Sanchez 18]. While examining previous grippers and the skills required for fabric manipulation in Chapter 2, a research gap was discovered in which no

existing devices were applicable to generalised pick-and-place textile applications. In addition, the literature highlighted that simply grasping and holding fabric requires dexterous skills, including grasping strategies that exploit environmental constraints and the capability to modulate the grasp force.

This initial gap inspired the research of Chapter 3, which sought to understand the manipulation and engineering limitations of previous devices. Given the numerous examples of human-inspired gripper designs, this research describes a subset of grippers designed for fabric manipulation using custom taxonomies based on anthropomorphic hand poses [Feix 15] and dexterous actions [Bullock 12]. While previous research has examined grippers for fabric handling [Borràs 20], the examination of these devices through an anthropomorphic lens was a new approach which previous research had not considered. In addition to surveying anthropomorphic comparisons, the technical details and applications of the surveyed mechanisms were considered. Surveying previous grippers under this anthropomorphic and technical lens led to several insights and a scope of dexterous capabilities for fabric pick-and-place described in Section 3.4.5. A number of further insights surrounding fabric handling beyond pick-and-place applications were also present, which Section 6.2 discusses.

The takeaways from the survey of previous devices informed the choice of a conceptual gripper based on the lateral grasp. Before proceeding with the design, construction and development of the gripper, Chapter 3 validated that the lateral grasp would be sufficient for pick-and-place with a user study outlined in Section 3.5. This research developed in Chapter 3 comprehensively responded to the first research question, laying the groundwork for the engineering undertaken in Chapter 4, which translated the conceptual lateral-grasp design into a realised robotic gripper, thereby addressing the second research question. The work undertaken in Chapter 4 also led to a publication [Hinwood 20]. The development of this device involved kinematic, kinostatic and dynamic analyses alongside an iterative prototyping process. Chapter 4 also evaluates the constructed manipulator's grasp strength and ability to exploit environmental constraints while grasping flattened clothing with pre-programmed trajectories.

Chapter 5 addresses the third and final research question. Using environmentally constrained (EC) grasping to pick up flattened clothing is a contact-rich manipulation task, which previous research performed with compliant mechanical elements [Koustoumpardis 14, Donaire 20]. Such approaches limited the range of wrist orientations in which this grasping act could occur, and also allowed friction to impact grasp success. To overcome these shortcomings, off-policy actor-critic reinforcement learning (RL) algorithms were implemented and deployed on a custom environment designed to encourage the gripper to perform EC grasping motions. The formulated reward signal used observations from a human-centric

study by Eppner *et al.* [Eppner 15], which saw how humans naturally exploit the environment when their vision is limited. Answering these three research questions followed a core tenet established in the literature, that human beings remain the most capable and diverse manipulators of fabric. By observing their behaviour and morphology, this thesis formulated, created, and trained a generalised robotic gripper for pick-and-place sorting of textile waste.

6.1.2 Limitations and Recommendations

Each chapter of research focused on a specific discipline within the broad field of robotics and contained unique limitations requiring diverse solutions. The anthropomorphic survey of Chapter 3, while effective in discovering the lateral grasp design inspiration, contained some limitations in its scope and applicability. Eight manipulators from the literature were chosen based on specific criteria that made them suitable for the anthropomorphic survey, as outlined in Section 3.2. The capabilities and features of this subset informed the custom taxonomies used, shown in Figures 3.11 and 3.14. This approach sufficiently outlined the skills required for generalised pick-and-place of fabric. However, exploring fabric manipulation in a broader context would require an expanded anthropomorphic survey investigating grippers with technical augmentations and commercial grippers while considering more complex manipulation primitives. Such an approach may extend the custom taxonomies built and derive further insights beyond the conclusions of Chapter 3. The survey of Borràs *et al.* [Borràs 20] is an example of such a comprehensive investigation, except they examined fabric manipulation under an object-centric framework. Finally, the human-centric survey conducted in Section 3.5 had a limited number of participants (17) requiring non-parametric statistical evaluations. By including a wider range of participants in the research, the conclusions drawn would gain increased robustness.

The modelling, construction and evaluation of Chapter 4 resulted in a unique gripper ready for data-driven experimentation. However, there were elements throughout this process that could have seen improvement. For example, the sensing setup on the hand will require refinement. At the gripper's present stage of development, a silicon tube with a pressure sensor detects rigid object collisions with the static plate by detecting changes in pressure when the tube deforms. This component must continuously detect forces applied to the static plate for a long-term solution. In addition, a different fingertip sensor may be required. Grasping and holding clothing may require a more biomimetic design that can detect collisions across the entire surface area of the 'fingertip'. While the applied sensor contributed to learning EC grasping in Chapter 5, the limited surface area and certain non-linear behaviours presented hurdles

that required addressing.

Finally, a slight discrepancy between the applied grasp force and the expected value from modelling was noted when evaluating the gripper's grasp strength ability. Despite this observed disparity, the gripper still exhibited a grasp strength appropriate for generalised garment handling and further investigation was not required to pursue the reinforcement learning in Chapter 5. Additionally, to go further with this gripper, an upgraded design more suited to commercial or long-term endeavours is required, in which case, further investigation expanding upon the design and modelling may yield further insights.

Lastly, the RL learning undertaken in Chapter 5 used two well-known off-policy actor-critic algorithms. This choice was made as previous robotic prototypes using similar actuators saw improved sample efficiency while learning collision-rich skills [Ibarz 21, Ahn 20]. The assumption was made that these algorithms would most effectively apply to the *arbitrary* grasping learning problem addressed in Chapter 5. However, given SAC's failure to converge and the time-intensive training process of TD3, further exploring the problem of environmentally constrained grasping with on-policy approaches such as Proximal Policy Optimisation (PPO) may be beneficial. In addition, despite failing to discover a set of hyperparameters allowing SAC to succeed at the RL of EC grasping with the gripper, further exploration with hyperparameter tuning or code-level optimisations could lead to SAC successfully learning in the simulation developed in Chapter 5 or both SAC and TD3 reducing their learning time.

6.2 Future Research Directions

A number of research avenues could build upon the research presented in this thesis. Regarding the anthropomorphic survey and human-centric study of Chapter 3, broadening the scope of manipulation beyond pick-and-place could yield a number of further insights for generalised fabric manipulation. In addition, growing the scope of grippers examined may also lead to valuable insights. One observation from the anthropomorphic survey saw that grippers considering folding challenges would perform environmentally constrained grasps that slide a compliant, thin finger appendage underneath a flattened garment, thereby preserving the garment's state. By exploring more complex tasks such as hanging up a shirt, folding garments, or spreading out garments under an anthropomorphic lens, more refined gripper designs capable of general garment handling could be realised.

Observations from the human-centric study in Chapter 3 reinforced previous notions from the literature, mainly that the act of folding remains a complex endeavour usually requiring more than two

fingers. However, this study also saw humans adopt a range of more complex manipulation primitives while handling clothing, such as draping clothes over their arms and torsos or adopting bimanual manipulation strategies under challenging situations. Investigating these more complex behaviours and when they occur could also lead to further insights for broader fabric manipulation development.

Refining the grippers design is a prominent aspect to explore in the near future. The ‘fingertip’ sensor on the gripper was the appropriate device to purchase during development due to financial and timeline obligations under the Ph.D. program. However, for generalised pick-and-place sorting of textile waste, a sensing configuration that can sense across a wide surface area of the TCP may improve the learning of EC grasping motions. One approach could involve revisiting the design thinking framework, emphasising sensing requirements throughout pick-and-place actions involving textiles. Furthermore, building upon the modelling process undertaken in Chapter 4, while investigating lower-level technical details could lead to an improved understanding of actuator behaviour and resolve discrepancies seen in Chapter 4 while investigating the grasp force. In addition to improving the modelling, further research on the gripper will investigate the in-hand dexterous skills of grasp gating, haptic exploration and slippage reaction.

Finally, expanding the deep reinforcement learning approaches of Chapter 5 could enable robots to perform various refined dexterous manipulation primitives. Immediate improvements to the present setup could include expanding the hyperparameters search and attempting to train on-policy algorithms, as mentioned in Section 6.1.2. In addition to these initial refinements, a goal-conditioned learning formulation using *Hindsight Experience Replay (HER, [Andrychowicz 17])* may allow policies to learn EC grasping motions that can modulate the wrench exerted at the TCP while dragging along the environment. Improved simulated environments that integrate fabrics or improve the domain randomisation aspects may also benefit learned policies. Alternatively, a multimodal setup that considers vision in combination with a gripper observation vector could also advantage future learning studies into EC grasping.

6.3 Final Remarks

Autonomous garment handling still requires further investigation in various engineering disciplines. Research focusing on gripper development, motion planning, computer vision, and deep learning will further improve robotic fabric manipulation. This thesis took inspiration from humans to build and train a serial-link manipulator with four actuators. Compared to previous devices, this gripper contains no

complex elements such as tendon-driven components or compliant joints, while remaining capable of the required skills of textile pick-and-place sorting. The human-centric perspective was the critical step to discovering this design and taking further inspiration from human manipulation skills and morphology could lead to improved approaches in fabric manipulation and broader autonomous interactions with the world.

From a personal perspective, the multidisciplinary skills learned throughout this Ph.D. program are valuable assets that will carry through to a broad range of future robotic endeavours. In addition to skills learned and experience gathered personally, this project has grown from research into a commercial endeavour. The company RediRobots was formed in 2022 and has received several government grants, alongside being accepted into an accelerator program to build a commercial version of the research presented throughout this thesis. As of August 2023, RediRobots is testing their textile sorting robots with recyclers throughout Australia, and I believe we can significantly impact the textile waste issue the world is experiencing today. This real-world impact would not have been possible without the research of this thesis, and I am grateful to my collaborators, supervisors, friends and mentors for their support throughout this journey.

Bibliography

- [Abbasi 16] Bahareh Abbasi, Ehsan Noohi, Sina Parastegari, and Miloš Žefran. *Grasp taxonomy based on force distribution*. In 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pages 1098–1103. IEEE, 2016.
- [Abe 20] Taiki Abe, Yuichi Kawasaki, and Kimitoshi Yamazaki. *A robotic end-effector with rolling up mechanism for pick-and-release of a cotton sheet*. ROBOMECH Journal, 7(1):1–9, 2020.
- [Ahn 20] 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.
- [Akkaya 19] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. *Solving rubik’s cube with a robot hand*. arXiv preprint arXiv:1910.07113, 2019.
- [Andrychowicz 17] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. *Hindsight experience replay*. arXiv preprint arXiv:1707.01495, 2017.
- [Arapi 21] Visar Arapi, Cosimo Della Santina, Giuseppe Averta, Antonio Bicchi, and Matteo Bianchi. *Understanding human manipulation with the environment: a novel taxonomy for video labelling*. IEEE Robotics and Automation Letters, 6(4):6537–6544, 2021.
- [Arbib 85] Michael Anthony Arbib, Thea Iberall, and Damian Lyons. *Coordinated control programs for movements of the hand*. Experimental brain research, pages 111–129, 1985.
- [Babin 21] Vincent Babin and Clément Gosselin. *Mechanisms for Robotic Grasping and Manipulation*. Annual Review of Control, Robotics, and Autonomous Systems, 4:573–593, 2021.

- [Balaguer 11] Benjamin Balaguer and Stefano Carpin. *Combining imitation and reinforcement learning to fold deformable planar objects*. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1405–1412. IEEE, 2011.
- [Bellicoso 15] Carmine Dario Bellicoso, Luca Rosario Buonocore, Vincenzo Lippiello, and Bruno Siciliano. *Design, modeling and control of a 5-DoF light-weight robot arm for aerial manipulation*. In 2015 23rd Mediterranean Conference on Control and Automation (MED), pages 853–858. IEEE, 2015.
- [Beltran-Hernandez 20] Cristian Camilo Beltran-Hernandez, Damien Petit, Ixchel Georgina Ramirez-Alpizar, Takayuki Nishi, Shinichi Kikuchi, Takamitsu Matsubara, and Kensuke Harada. *Learning force control for contact-rich manipulation tasks with rigid position-controlled robots*. IEEE Robotics and Automation Letters, 5(4):5709–5716, 2020.
- [Besser 21] Linton Besser. *‘Dead white man’s clothes’: The dirty secret behind the world’s fashion addiction*, Aug 2021.
- [Bicchi 00] Antonio Bicchi and Vijay Kumar. *Robotic grasping and contact: A review*. In Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), Volume 1, pages 348–353. IEEE, 2000.
- [Bick 18] Rachel Bick, Erika Halsey, and Christine C Ekenga. *The global environmental injustice of fast fashion*. Environmental Health, 17(1):1–4, 2018.
- [Billard 19] Aude Billard and Danica Kragic. *Trends and challenges in robot manipulation*. Science, 364(6446), 2019.
- [Bjelonic 18] Marko Bjelonic, Navinda Kottege, Timon Homberger, Paulo Borges, Philipp Beckerle, and Margarita Chli. *Weaver: Hexapod robot for autonomous navigation on unstructured terrain*. Journal of Field Robotics, 35(7):1063–1079, 2018.
- [Borràs 20] Júlia Borràs, Guillem Alenyà, and Carme Torras. *A grasping-centered analysis for cloth manipulation*. IEEE Transactions on Robotics, 36(3):924–936, 2020.
- [Brockman 16] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. *OpenAI Gym*, 2016.

- [Bullock 11] Ian M Bullock and Aaron M Dollar. *Classifying human manipulation behavior*. In 2011 IEEE International Conference on Rehabilitation Robotics, pages 1–6. IEEE, 2011.
- [Bullock 12] Ian M Bullock, Raymond R Ma, and Aaron M Dollar. *A hand-centric classification of human and robot dexterous manipulation*. IEEE Transactions on Haptics, 6(2):129–144, 2012.
- [Bullock 13] Ian M Bullock, Joshua Z Zheng, Sara De La Rosa, Charlotte Guertler, and Aaron M Dollar. *Grasp frequency and usage in daily household and machine shop tasks*. IEEE Transactions on Haptics, 6(3):296–308, 2013.
- [Bullock 15] Ian M Bullock, Thomas Feix, and Aaron M Dollar. *The Yale human grasping dataset: Grasp, object, and task data in household and machine shop environments*. The International Journal of Robotics Research, 34(3):251–255, 2015.
- [Catalano 14] Manuel G Catalano, Giorgio Grioli, Edoardo Farnioli, Alessandro Serio, Cristina Piazza, and Antonio Bicchi. *Adaptive synergies for the design and control of the Pisa/IIT SoftHand*. The International Journal of Robotics Research, 33(5):768–782, 2014.
- [Chan 19] Stephanie CY Chan, Samuel Fishman, John Canny, Anoop Korattikara, and Sergio Guadarrama. *Measuring the reliability of reinforcement learning algorithms*. arXiv preprint arXiv:1912.05663, 2019.
- [Choi 17] Myoung-Su Choi, Dong-Hyuk Lee, Hyeonjun Park, Young-Jin Kim, Ga-Ram Jang, Yong-Deuk Shin, Jae-Han Park, Moon-Hong Baeg, and Ji-Hun Bae. *Development of multi-purpose universal gripper*. In 2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), pages 1421–1424. IEEE, 2017.
- [Choi 18] Changhyun Choi, Wilko Schwarting, Joseph DelPreto, and Daniela Rus. *Learning object grasping for soft robot hands*. IEEE Robotics and Automation Letters, 3(3):2370–2377, 2018.
- [Chung 16] Se-Joon Chung and Nancy Pollard. *Predictable behavior during contact simulation: a comparison of selected physics engines*. Computer Animation and Virtual Worlds, 27(3-4):262–270, 2016.
- [Church 22] Alex Church, John Lloyd, Nathan F Lepora, et al. *Tactile sim-to-real policy transfer via real-to-sim image translation*. In Conference on Robot Learning, pages 1645–1654. PMLR, 2022.

- [Claudio 07] Luz Claudio. *Waste couture: Environmental impact of the clothing industry*, 2007.
- [Collins 19] Jack Collins, David Howard, and Jurgen Leitner. *Quantifying the reality gap in robotic manipulation tasks*. In 2019 International Conference on Robotics and Automation (ICRA), pages 6706–6712. IEEE, 2019.
- [Collins 20] Jack Collins, Ross Brown, Jurgen Leitner, and David Howard. *Traversing the reality gap via simulator tuning*. arXiv preprint arXiv:2003.01369, 2020.
- [Collins 21] Jack Collins, Shelvin Chand, Anthony Vanderkop, and David Howard. *A Review of Physics Simulators for Robotic Applications*. IEEE Access, 2021.
- [Corke 96] Peter I Corke. *A robotics toolbox for MATLAB*. IEEE Robotics & Automation Magazine, 3(1):24–32, 1996.
- [Corke 17] Peter Corke. *Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised*, Volume 118. Springer, 2017.
- [Coumans 16] E Coumans and Y Bai. *Pybullet, a python module for physics simulation for games*. Robotics and machine learning.[Google Scholar], 2016.
- [Cutkosky 89] Mark R Cutkosky et al. *On grasp choice, grasp models, and the design of hands for manufacturing tasks*. IEEE Transactions on robotics and automation, 5(3):269–279, 1989.
- [Dafle 14] 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.
- [Deimel 16] 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.
- [Della Santina 17] Cosimo Della Santina, Matteo Bianchi, Giuseppe Averta, Simone Ciotti, Visar Arapi, Simone Fani, Edoardo Battaglia, Manuel Giuseppe Catalano, Marco Santello, and Antonio Bicchi. *Postural hand synergies during environmental constraint exploitation*. Frontiers in Neurorobotics, 11:41, 2017.

- [Denei 17] Simone Denei, Perla Maiolino, Emanuele Baglini, and Giorgio Cannata. *Development of an integrated tactile sensor system for clothes manipulation and classification using industrial grippers*. IEEE Sensors Journal, 17(19):6385–6396, 2017.
- [Donaire 20] Sonia Donaire, Júlia Borrás, Guillem Alenya, and Carme Torras. *A versatile gripper for cloth manipulation*. IEEE Robotics and Automation Letters, 5(4):6520–6527, 2020.
- [Dragusanu 22] Mihai Dragusanu, Sara Marullo, Monica Malvezzi, Gabriele Maria Achilli, Maria Cristina Valigi, Domenico Prattichizzo, and Gionata Salviati. *The dressgripper: A collaborative gripper with electromagnetic fingertips for dressing assistance*. IEEE Robotics and Automation Letters, 7(3):7479–7486, 2022.
- [Elguea-Aguinaco 23] Íñigo Elguea-Aguinaco, Antonio Serrano-Muñoz, Dimitrios Chrysostomou, Ibai Inziarte-Hidalgo, Simon Bøgh, and Nestor Arana-Arexolaleiba. *A review on reinforcement learning for contact-rich robotic manipulation tasks*. Robotics and Computer-Integrated Manufacturing, 81:102517, 2023.
- [Engstrom 19] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. *Implementation matters in deep rl: A case study on ppo and trpo*. In International conference on learning representations, 2019.
- [EPA 21] EPA. *Advancing Sustainable Materials Management: Facts and Figures Report*, 2021.
- [Eppner 13] Clemens Eppner and Oliver Brock. *Grasping unknown objects by exploiting shape adaptability and environmental constraints*. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4000–4006. IEEE, 2013.
- [Eppner 15] Clemens Eppner, Raphael Deimel, José Alvarez-Ruiz, Marianne Maertens, and Oliver Brock. *Exploitation of environmental constraints in human and robotic grasping*. The International Journal of Robotics Research, 34(7):1021–1038, 2015.
- [Erez 15] Tom Erez, Yuval Tassa, and Emanuel Todorov. *Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx*. In 2015 IEEE international conference on robotics and automation (ICRA), pages 4397–4404. IEEE, 2015.

- [Faigl 19] Jan Faigl and Petr Čížek. *Adaptive locomotion control of hexapod walking robot for traversing rough terrains with position feedback only*. Robotics and Autonomous Systems, 116:136–147, 2019.
- [Feix 09] Thomas Feix, Roland Pawlik, Heinz-Bodo Schmiedmayer, Javier Romero, and Danica Kragic. *A comprehensive grasp taxonomy*. In Robotics, science and systems: workshop on understanding the human hand for advancing robotic manipulation, Volume 2, pages 2–3. Seattle, WA, USA, 2009.
- [Feix 14] Thomas Feix, Ian M Bullock, and Aaron M Dollar. *Analysis of human grasping behavior: Correlating tasks, objects and grasps*. IEEE Transactions on Haptics, 7(4):430–441, 2014.
- [Feix 15] 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.
- [Fujimoto 18] Scott Fujimoto, Herke Hoof, and David Meger. *Addressing function approximation error in actor-critic methods*. In International Conference on Machine Learning, pages 1587–1596. PMLR, 2018.
- [Ganapathi 22] Aditya Ganapathi, Pete Florence, Jake Varley, Kaylee Burns, Ken Goldberg, and Andy Zeng. *Implicit Kinematic Policies: Unifying Joint and Cartesian Action Spaces in End-to-End Robot Learning*. arXiv preprint arXiv:2203.01983, 2022.
- [Gautier 13] Maxime Gautier and Gentiane Venture. *Identification of standard dynamic parameters of robots with positive definite inertia matrix*. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 5815–5820. IEEE, 2013.
- [Gosselin 08] Clement Gosselin, Frederic Pelletier, and Thierry Laliberte. *An anthropomorphic underactuated robotic hand with 15 dofs and a single actuator*. In 2008 IEEE International Conference on Robotics and Automation, pages 749–754. IEEE, 2008.
- [Gu 17] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. *Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates*. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 3389–3396. IEEE, 2017.

- [Gupta 21] Abhishek Gupta, Justin Yu, Tony Z Zhao, Vikash Kumar, Aaron Rovinsky, Kelvin Xu, Thomas Devlin, and Sergey Levine. *Reset-free reinforcement learning via multi-task learning: Learning dexterous manipulation behaviors without human intervention*. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 6664–6671. IEEE, 2021.
- [Haarnoja 17] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. *Reinforcement learning with deep energy-based policies*. In International Conference on Machine Learning, pages 1352–1361. PMLR, 2017.
- [Haarnoja 18a] Tuomas Haarnoja, Sehoon Ha, Aurick Zhou, Jie Tan, George Tucker, and Sergey Levine. *Learning to walk via deep reinforcement learning*. arXiv preprint arXiv:1812.11103, 2018.
- [Haarnoja 18b] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. *Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor*. In International conference on machine learning, pages 1861–1870. PMLR, 2018.
- [Haarnoja 18c] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. *Soft actor-critic algorithms and applications*. arXiv preprint arXiv:1812.05905, 2018.
- [Han 23] Dong Han, Beni Mulyana, Vladimir Stankovic, and Samuel Cheng. *A Survey on Deep Reinforcement Learning Algorithms for Robotic Manipulation*. *Sensors*, 23(7):3762, 2023.
- [Hansen 22] Nicklas Hansen, Xiaolong Wang, and Hao Su. *Temporal Difference Learning for Model Predictive Control*. arXiv preprint arXiv:2203.04955, 2022.
- [Heikkilä 19] Pirjo Heikkilä, Kirsti Cura, Jouko Heikkilä, Ville Hinkka, Tiina Ikonen, Taina Kamppuri, Henna Knuutila, Milja Kokko, Sonja Lankiniemi, Liisa Lehtinen, et al. *Telaketju: Towards circularity of textiles*. VTT Research Information Portal, 2019.
- [Heinemann 15] Fabian Heinemann, Steffen Puhmann, Clemens Eppner, José Élvarez-Ruiz, Marianne Maertens, and Oliver Brock. *A taxonomy of human grasping behavior suitable for transfer to robotic hands*. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 4286–4291. IEEE, 2015.

- [Henderson 18] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. *Deep reinforcement learning that matters*. In Proceedings of the AAAI conference on artificial intelligence, Volume 32, 2018.
- [Herath 22] Damith Herath and David St-Onge. *Foundations of Robotics: A Multidisciplinary Approach with Python and ROS*. 2022.
- [Hinwood 20] David Hinwood, Damith Herath, and Roland Goecke. *Towards the Design of a Human-Inspired Gripper for Textile Manipulation*. In IEEE International Conference on Automation Science and Engineering (CASE), 2020.
- [Huang 21] Wenlong Huang, Igor Mordatch, Pieter Abbeel, and Deepak Pathak. *Generalization in Dexterous Manipulation via Geometry-Aware Multi-Task Learning*. arXiv preprint arXiv:2111.03062, 2021.
- [Ibarz 21] Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. *How to train your robot with deep reinforcement learning: lessons we have learned*. The International Journal of Robotics Research, 40(4-5):698–721, 2021.
- [Iberall 86] Thea Iberall. *Opposition space as a structuring concept for the analysis of skilled hand movements*. Generation and modulation of action patterns, 15:158–173, 1986.
- [Iberall 97] Thea Iberall. *Human prehension and dexterous robot hands*. The International Journal of Robotics Research, 16(3):285–299, 1997.
- [Inc 21] ROBOTIS Inc. *ROBOTIS e-manual - XM-430-350 T/R*, 2021.
- [Ioffe 15] Sergey Ioffe and Christian Szegedy. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. In International conference on machine learning, pages 448–456. PMLR, 2015.
- [Jiménez 17] Pablo Jiménez. *Visual grasp point localization, classification and state recognition in robotic manipulation of cloth: An overview*. Robotics and Autonomous Systems, 92:107–125, 2017.
- [Joung 13] Hyun-Mee Joung and Haesun Park-Poaps. *Factors motivating and influencing clothing disposal behaviours*. International Journal of consumer studies, 37(1):105–111, 2013.

- [Kaboli 16] Mohsen Kaboli, Kunpeng Yao, and Gordon Cheng. *Tactile-based manipulation of deformable objects with dynamic center of mass*. In 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), pages 752–757. IEEE, 2016.
- [Kalashnikov 18] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. *Scalable deep reinforcement learning for vision-based robotic manipulation*. In Conference on Robot Learning, pages 651–673. PMLR, 2018.
- [Kamakura 80] Noriko Kamakura, Michiko Matsuo, Harumi Ishii, Fumiko Mitsuboshi, and Yoriko Miura. *Patterns of static prehension in normal hands*. *American Journal of Occupational Therapy*, 34(7):437–445, 1980.
- [Kang 92] Sing Bing Kang and Katsushi Ikeuchi. *Grasp Recognition Using The Contact Web*. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 194–201, 1992.
- [Kant 11] Rita Kant. *Textile dyeing industry an environmental hazard*. *Natural Science*, 2011.
- [Kaspar 20] Manuel Kaspar, Juan D Muñoz Osorio, and Jürgen Bock. *Sim2real transfer for reinforcement learning without dynamics randomization*. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4383–4388. IEEE, 2020.
- [Kazemi 14] Moslem Kazemi, Jean-Sebastien Valois, J Andrew Bagnell, and Nancy Pollard. *Human-inspired force compliant grasping primitives*. *Autonomous Robots*, 37(2):209–225, 2014.
- [Khamis 18] Heba Khamis, Raquel Izquierdo Albero, Matteo Salerno, Ahmad Shah Idil, Andrew Loizou, and Stephen J Redmond. *Papillary: An incipient slip sensor for dexterous robotic or prosthetic manipulation—design and prototype validation*. *Sensors and Actuators A: Physical*, 270:195–204, 2018.
- [Khamis 19] Heba Khamis, Benjamin Xia, and Stephen J Redmond. *A novel optical 3D force and displacement sensor—Towards instrumenting the PapillArray tactile sensor*. *Sensors and Actuators A: Physical*, 291:174–187, 2019.

- [Khamis 21] Heba Khamis, Benjamin Xia, and Stephen J Redmond. *Real-time Friction Estimation for Grip Force Control*. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 1608–1614. IEEE, 2021.
- [Kingma 14] Diederik P Kingma and Jimmy Ba. *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980, 2014.
- [Kita 11] Yasuyo Kita, Fumio Kanehiro, Toshio Ueshiba, and Nobuyuki Kita. *Clothes handling based on recognition by strategic observation*. In 2011 11th IEEE-RAS International Conference on Humanoid Robots, pages 53–58. IEEE, 2011.
- [Körber 21] Marian Körber, Johann Lange, Stephan Rediske, Simon Steinmann, and Roland Glück. *Comparing Popular Simulation Environments in the Scope of Robotics and Reinforcement Learning*. arXiv preprint arXiv:2103.04616, 2021.
- [Koustoumpardis 04] PN Koustoumpardis and NA Aspragathos. *A review of gripping devices for fabric handling*. *hand*, 19:20, 2004.
- [Koustoumpardis 14] Panagiotis N Koustoumpardis, Kostas X Nastos, and Nikos A Aspragathos. *Underactuated 3-finger robotic gripper for grasping fabrics*. In 2014 23rd International Conference on Robotics in Alpe-Adria-Danube Region (RAAD), pages 1–8. IEEE, 2014.
- [Koustoumpardis 17] Panagiotis N Koustoumpardis, Sotiris Smyrnis, and Nikos A Aspragathos. *A 3-finger robotic gripper for grasping fabrics based on CAMS-followers mechanism*. In International Conference on Robotics in Alpe-Adria Danube Region, pages 612–620. Springer, 2017.
- [Ku 20] Subyeong Ku, Jihye Myeong, Ho-Young Kim, and Yong-Lae Park. *Delicate fabric handling using a soft robotic gripper with embedded microneedles*. *IEEE Robotics and Automation Letters*, 5(3):4852–4858, 2020.
- [Le 13] Loan Le, Matteo Zoppi, Michal Jilich, Raffaello Camoriano, Dimiter Zlatanov, and Rezia Molfino. *Development and analysis of a new specialized gripper mechanism for garment handling*. In ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pages V06BT07A013–V06BT07A013. American Society of Mechanical Engineers, 2013.

- [Le 15a] Loan Le, Matteo Zoppi, Michal Jilich, Han Bo, Dimiter Zlatanov, and Rezia Molfino. *Application of a biphasic actuator in the design of the CloPeMa robot gripper*. *Journal of Mechanisms and Robotics*, 7(1):011011, 2015.
- [Le 15b] Loan Le, Matteo Zoppi, Michal Jilich, Han Bo, Dimiter Zlatanov, and Rezia Molfino. *Application of a biphasic actuator in the design of the CloPeMa robot gripper*. *Journal of Mechanisms and Robotics*, 7(1):11011, 2015.
- [Lee 21] Myoung Hoon Lee and Jun Moon. *Deep Reinforcement Learning-based UAV Navigation and Control: A Soft Actor-Critic with Hindsight Experience Replay Approach*. arXiv preprint arXiv:2106.01016, 2021.
- [Lewrick 18] Michael Lewrick, Patrick Link, and Larry Leifer. *The design thinking playbook: Mindful digital transformation of teams, products, services, businesses and ecosystems*. John Wiley & Sons, 2018.
- [Li 15] Yinxiao Li, Yonghao Yue, Danfei Xu, Eitan Grinspun, and Peter K Allen. *Folding deformable objects using predictive simulation and trajectory optimization*. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 6000–6006. IEEE, 2015.
- [Lillicrap 15] 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*. arXiv preprint arXiv:1509.02971, 2015.
- [Liu 14] 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. IEEE, 2014.
- [Liu 22] Marc Liu. *Time to make fast fashion a problem for its makers, not charities*. *The Conversation*, May 2022.
- [Luo 18] Shan Luo, Wenzhen Yuan, Edward Adelson, Anthony G Cohn, and Raul Fuentes. *Vitac: Feature sharing between vision and tactile sensing for cloth texture recognition*. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 2722–2727. IEEE, 2018.

- [Ma 16] Raymond R Ma, Adam Spiers, and Aaron M Dollar. *M 2 gripper: Extending the dexterity of a simple, underactuated gripper*. In Advances in reconfigurable mechanisms and robots II, pages 795–805. Springer, 2016.
- [Ma 17] Raymond Ma and Aaron Dollar. *Yale openhand project: Optimizing open-source hand designs for ease of fabrication and adoption*. IEEE Robotics & Automation Magazine, 24(1):32–40, 2017.
- [Ma 21] Li-Ke Ma, Zeshi Yang, Tong Xin, Baining Guo, and KangKang Yin. *Learning and Exploring Motor Skills with Spacetime Bounds*. Computer Graphics Forum, 40(2), 2021.
- [Maitin-Shepard 10] Jeremy Maitin-Shepard, Marco Cusumano-Towner, Jinna Lei, and Pieter Abbeel. *Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding*. In 2010 IEEE International Conference on Robotics and Automation, pages 2308–2315. IEEE, 2010.
- [Marullo 20] Sara Marullo, Simone Bartoccini, Gionata Salvietti, Muhammad Zubair Iqbal, and Domenico Prattichizzo. *The mag-gripper: A soft-rigid gripper augmented with an electro-magnet to precisely handle clothes*. IEEE Robotics and Automation Letters, 5(4):6591–6598, 2020.
- [Matas 18] Jan Matas, Stephen James, and Andrew J Davison. *Sim-to-real reinforcement learning for deformable object manipulation*. In Conference on Robot Learning, pages 734–743. PMLR, 2018.
- [Meng 21] Lingheng Meng, Rob Gorbet, and Dana Kulić. *Memory-based Deep Reinforcement Learning for POMDP*. arXiv preprint arXiv:2102.12344, 2021.
- [Mnih 13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. *Playing atari with deep reinforcement learning*. arXiv preprint arXiv:1312.5602, 2013.
- [Monsó 12] Pol Monsó, Guillem Alenyà, and Carme Torras. *Pomdp approach to robotized clothes separation*. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1324–1329. IEEE, 2012.

- [Moriya 18] Yusuke Moriya, Daisuke Tanaka, Kimitoshi Yamazaki, and Keisuke Takeshita. *A method of picking up a folded fabric product by a single-armed robot*. ROBOMECH Journal, 5(1):1, 2018.
- [Murakami 04] Kouji Murakami and Tsutomu Hasegawa. *Novel fingertip equipped with soft skin and hard nail for dexterous multi-fingered robotic manipulation*. Journal of the Robotics Society of Japan, 22(5):616–624, 2004.
- [Mutlu 15] Rahim Mutlu, Gursel Alici, Marc in het Panhuis, and Geoff Spinks. *Effect of flexure hinge type on a 3D printed fully compliant prosthetic finger*. In 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), pages 790–795. IEEE, 2015.
- [Nakamura 17] Yuzuko C Nakamura, Daniel M Troniak, Alberto Rodriguez, Matthew T Mason, and Nancy S Pollard. *The complexities of grasping in the wild*. In 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), pages 233–240. IEEE, 2017.
- [Napier 56] John R Napier. *The prehensile movements of the human hand*. The Journal of bone and joint surgery. British volume, 38(4):902–913, 1956.
- [Newbury 23] Rhys Newbury, Morris Gu, Lachlan Chumbley, Arsalan Mousavian, Clemens Eppner, Jürgen Leitner, Jeannette Bohg, Antonio Morales, Tamim Asfour, Danica Kragic, et al. *Deep learning approaches to grasp synthesis: A review*. IEEE Transactions on Robotics, 2023.
- [Nielsen 99] James Nielsen and Bernard Roth. *On the kinematic analysis of robotic mechanisms*. The International Journal of Robotics Research, 18(12):1147–1160, 1999.
- [Niinimäki 20] Kirsi Niinimäki, Greg Peters, Helena Dahlbo, Patsy Perry, Timo Rissanen, and Alison Gwilt. *The environmental price of fast fashion*. Nature Reviews Earth & Environment, 1(4):189–200, 2020.
- [Odhner 12] Lael U Odhner, Raymond R Ma, and Aaron M Dollar. *Precision grasping and manipulation of small objects from flat surfaces using underactuated fingers*. In 2012 IEEE International Conference on Robotics and Automation, pages 2830–2835. IEEE, 2012.
- [Odhner 14] Lael U Odhner, Leif P Jentoft, Mark R Claffee, Nicholas Corson, Yaroslav Tenzer, Raymond R Ma, Martin Buehler, Robert Kohout, Robert D Howe, and Aaron M Dollar. *A com-*

- pliant, underactuated hand for robust manipulation*. The International Journal of Robotics Research, 33(5):736–752, 2014.
- [Ono 91] Eiichi Ono, Hishao Ichijou, and Noborou Aisaka. *Robot hand for handling cloth*. In Fifth International Conference on Advanced Robotics' Robots in Unstructured Environments, pages 769–774. IEEE, 1991.
- [Ono 01] Eiichi Ono, Kosei Kitagaki, and Masayoshi Kakikura. *Picking up a piece of fabric from layers by a hand with 3 fingers and a palm*. In Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180), Volume 2, pages 931–936. IEEE, 2001.
- [Ono 05] Eiichi Ono, Kosei Kitagaki, and Masayoshi Kakikura. *On friction picking up a piece of fabric from layers*. In IEEE International Conference Mechatronics and Automation, 2005, Volume 4, pages 2206–2211. IEEE, 2005.
- [Ono 07] Eiichi Ono and Kunikatsu Takase. *On better pushing for picking a piece of fabric from layers*. In 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO), pages 589–594. IEEE, 2007.
- [Parker 83] JK Parker, R Dubey, FW Paul, and RJ Becker. *Robotic Fabric Handling for Automating Garment Manufacturing*. Journal of Engineering for Industry, 105:21, 1983.
- [Peng 18] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. *Sim-to-real transfer of robotic control with dynamics randomization*. In 2018 IEEE international conference on robotics and automation (ICRA), pages 3803–3810. IEEE, 2018.
- [Peng 20] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Edward Lee, Jie Tan, and Sergey Levine. *Learning Agile Robotic Locomotion Skills by Imitating Animals*. In Robotics: Science and Systems, 07 2020.
- [Pensupa 17] Nattha Pensupa, Shao-Yuan Leu, Yunzi Hu, Chenyu Du, Hao Liu, Houde Jing, Huaimin Wang, and Carol Sze Ki Lin. *Recent trends in sustainable textile waste recycling methods: Current situation and future prospects*. Chemistry and Chemical Technologies in Waste Valorization, pages 189–228, 2017.

- [Peters 06] Jan Peters and Stefan Schaal. *Policy gradient methods for robotics*. In 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2219–2225. IEEE, 2006.
- [Petřík 15] Vladimír Petřík, Vladimír Smutný, Pavel Krsek, and Václav Hlaváč. *Robotic garment folding: Precision improvement and workspace enlargement*. In Towards Autonomous Robotic Systems: 16th Annual Conference, TAROS 2015, Liverpool, UK, September 8-10, 2015, Proceedings 16, pages 204–215. Springer, 2015.
- [Puhlmann 16] Steffen Puhlmann, Fabian Heinemann, Oliver Brock, and Marianne Maertens. *A compact representation of human single-object grasping*. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1954–1959. IEEE, 2016.
- [Rajeswaran 17] 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.
- [Rao 20] Kanishka Rao, Chris Harris, Alex Irpan, Sergey Levine, Julian Ibarz, and Mohi Khansari. *RL-cyclegan: Reinforcement learning aware simulation-to-real*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11157–11166, 2020.
- [Roby-Brami 03] Agnès Roby-Brami, Stéphane Jacobs, Nezha Bennis, and Mindy F Levin. *Hand orientation for grasping and arm joint rotation patterns in healthy subjects and hemiparetic stroke patients*. Brain research, 969(1-2):217–229, 2003.
- [Ruan 18] Mengyao Ruan, Dale Mc Conachie, and Dmitry Berenson. *Accounting for Directional Rigidity and Constraints in Control for Manipulation of Deformable Objects without Physical Simulation*. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 512–519. IEEE, 2018.
- [Sahari 10] Khairul Salleh Mohamed Sahari, Hiroaki Seki, Yoshitsugu Kamiya, and Masatoshi Hikizu. *Edge tracing manipulation of clothes based on different gripper types*. Journal of Computer Science, 6(8):872–879, 2010.
- [Şahin 19] Murat Şahin and Eren Aybek. *Jamovi: an easy to use statistical software for the social scientists*. International Journal of Assessment Tools in Education, 6(4):670–692, 2019.

- [Salvietti 15] Gionata Salvietti, Monica Malvezzi, Guido Gioioso, and Domenico Prattichizzo. *Modeling compliant grasps exploiting environmental constraints*. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 4941–4946. IEEE, 2015.
- [Sanchez 18] Jose Sanchez, Juan-Antonio Corrales, Belhassen-Chedli Bouzgarrou, and Youcef Mezouar. *Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey*. The International Journal of Robotics Research, 37(7):688–716, 2018.
- [Sarantopoulos 18] Iason Sarantopoulos and Zoe Doulgeri. *Human-inspired robotic grasping of flat objects*. Robotics and Autonomous Systems, 108:179–191, 2018.
- [Schaul 15] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. *Prioritized experience replay*. arXiv preprint arXiv:1511.05952, 2015.
- [Schoettler 20] Gerrit Schoettler, Ashvin Nair, Jianlan Luo, Shikhar Bahl, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. *Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards*. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5548–5555. IEEE, 2020.
- [Schulman 15] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. *Trust region policy optimization*. In International Conference on Machine Learning, pages 1889–1897. PMLR, 2015.
- [Schulman 17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. *Proximal policy optimization algorithms*. arXiv preprint arXiv:1707.06347, 2017.
- [Sciavicco 12] Lorenzo Sciavicco and Bruno Siciliano. *Modelling and control of robot manipulators*. Springer Science & Business Media, 2012.
- [Seita 21] Daniel Seita, Pete Florence, Jonathan Tompson, Erwin Coumans, Vikas Sindhwani, Ken Goldberg, and Andy Zeng. *Learning to Rearrange Deformable Cables, Fabrics, and Bags with Goal-Conditioned Transporter Networks*. In IEEE International Conference on Robotics and Automation (ICRA), 2021.
- [Seminara 23] Lucia Seminara, Strahinja Dosen, Fulvio Mastrogiovanni, Matteo Bianchi, Simon Watt, Philipp Beckerle, Thrishantha Nanayakkara, Knut Drewing, Alessandro Moscatelli, Roberta L

- Klatzky, et al. *A hierarchical sensorimotor control framework for human-in-the-loop robotic hands*. *Science Robotics*, 8(78):eadd5434, 2023.
- [Shibata 08] Mizuho Shibata, Tsuyoshi Ota, Yoshimasa Endo, and Shinichi Hirai. *Handling of hemmed fabrics by a single-armed robot*. In 2008 IEEE International Conference on Automation Science and Engineering, pages 882–887. IEEE, 2008.
- [Shibata 09] Mizuho Shibata, Tsuyoshi Ota, and Shinichi Hirai. *Wiping motion for deformable object handling*. In 2009 IEEE International Conference on Robotics and Automation, pages 134–139. IEEE, 2009.
- [Shibata 12] Mizuho Shibata and Shinichi Hirai. *Fabric manipulation utilizing contacts with the environment*. In 2012 IEEE International Conference on Automation Science and Engineering (CASE), pages 442–447. IEEE, 2012.
- [Shibata 16] Mizuho Shibata and Shinichi Hirai. *A Pinching Strategy for Fabrics Using Wiping Deformation*. *Robotics*, 5(2):10, 2016.
- [Shui 11] S Shui and A Plastina. *A summary of the world apparel fiber consumption survey 2005–2008*. Food and Agriculture Organization of the United Nation (FAO) and International Cotton Advisory Committee (ICAC), Washington, pages 1–11, 2011.
- [Siegle 19] Lucy Siegle. *Fast fashion is on the rampage, with the UK at the head of the charge*. *The Guardian*, Jun 2019.
- [Sikchi 22] Harshit Sikchi, Wenxuan Zhou, and David Held. *Learning off-policy with online planning*. In Conference on Robot Learning, pages 1622–1633. PMLR, 2022.
- [Silver 14] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. *Deterministic policy gradient algorithms*. In International Conference on Machine Learning, pages 387–395. PMLR, 2014.
- [Singh 19] Avi Singh, Larry Yang, Kristian Hartikainen, Chelsea Finn, and Sergey Levine. *End-to-end robotic reinforcement learning without reward engineering*. arXiv preprint arXiv:1904.07854, 2019.
- [Singla 20] Abhik Singla, Shalabh Bhatnagar, et al. *Hindsight Experience Replay with Kronecker Product Approximate Curvature*. arXiv preprint arXiv:2010.06142, 2020.

- [Song 18] Doo Re Song, Chuanyu Yang, Christopher McGreavy, and Zhibin Li. *Recurrent deterministic policy gradient method for bipedal locomotion on rough terrain challenge*. In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), pages 311–318. IEEE, 2018.
- [Spiers 18] Adam J Spiers, Berk Calli, and Aaron M Dollar. *Variable-friction finger surfaces to enable within-hand manipulation via gripping and sliding*. IEEE Robotics and Automation Letters, 3(4):4116–4123, 2018.
- [Stival 19] Francesca Stival, Stefano Michieletto, Matteo Cognolato, Enrico Pagello, Henning Müller, and Manfredo Atzori. *A quantitative taxonomy of human hand grasps*. Journal of Neuroengineering and Rehabilitation, 16(1):1–17, 2019.
- [Suomalainen 22] Markku Suomalainen, Yiannis Karayiannidis, and Ville Kyrki. *A survey of robot manipulation in contact*. Robotics and Autonomous Systems, 156:104224, 2022.
- [Sutton 18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [Tan 18] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. *Sim-to-real: Learning agile locomotion for quadruped robots*. arXiv preprint arXiv:1804.10332, 2018.
- [Tassa 18] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. *Deepmind control suite*. arXiv preprint arXiv:1801.00690, 2018.
- [Thrun 93] Sebastian Thrun and Anton Schwartz. *Issues in using function approximation for reinforcement learning*. In Proceedings of the Fourth Connectionist Models Summer School, pages 255–263. Hillsdale, NJ, 1993.
- [Thuy-Hong-Loan Le 13] Michal Jilich Thuy-Hong-Loan Le, Alberto Landini, Matteo Zoppi, Dimiter Zlatanov, and Rezia Molfino. *On the development of a specialized flexible gripper for garment handling*. Journal of Automation and Control Engineering Vol, 1(3), 2013.

- [Trivun 17] Darko Trivun, Haris Dindo, and Bakir Lačević. *Resilient hexapod robot*. In 2017 XXVI International Conference on Information, Communication and Automation Technologies (ICAT), pages 1–6. IEEE, 2017.
- [Twardon 15] Lukas Twardon and Helge Ritter. *Interaction skills for a coat-check robot: Identifying and handling the boundary components of clothes*. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 3682–3688. IEEE, 2015.
- [Tyldesley 96] Barbara Tyldesley and June I Grieve. *Muscles, nerves, and movement: kinesiology in daily living*. Blackwell Science, 1996.
- [Varin 19] Patrick Varin, Lev Grossman, and Scott Kuindersma. *A comparison of action spaces for learning manipulation tasks*. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 6015–6021. IEEE, 2019.
- [Verleysen 20] Andreas Verleysen, Matthijs Biondina, and Francis Wyffels. *Video dataset of human demonstrations of folding clothing for robotic folding*. *The International Journal of Robotics Research*, 39(9):1031–1036, 2020.
- [Von Drigalski 17a] Felix Von Drigalski, Marcus Gall, Sung-Gwi Cho, Ming Ding, Jun Takamatsu, Tsukasa Ogasawara, and Tamim Asfour. *Textile identification using fingertip motion and 3D force sensors in an open-source gripper*. In 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), pages 424–429. IEEE, 2017.
- [Von Drigalski 17b] Felix Von Drigalski, Daiki Yoshioka, Wataru Yamazaki, Sung-Gwi Cho, Marcus Gall, Pedro Miguell Uriguen Eljuri, Viktor Hoerig, Ming Ding, Jun Takamatsu, Tsukasa Ogasawara, et al. *NAIST Openhand M2S: A Versatile Two-Finger Gripper Adapted for Pulling and Tucking Textile*. In 2017 First IEEE International Conference on Robotic Computing (IRC), pages 117–122. IEEE, 2017.
- [Wang 10] Youjiang Wang. *Fiber and textile waste utilization*. *Waste and Biomass Valorization*, 1(1):135–143, 2010.
- [Wang 20a] Cong Wang, Qifeng Zhang, Qiyang Tian, Shuo Li, Xiaohui Wang, David Lane, Yvan Petillot, and Sen Wang. *Learning mobile manipulation through deep reinforcement learning*. *Sensors*, 20(3):939, 2020.

- [Wang 20b] Yufei Wang and Tianwei Ni. *Meta-SAC: Auto-tune the Entropy Temperature of Soft Actor-Critic via Metagradient*. arXiv preprint arXiv:2007.01932, 2020.
- [Xiang 21] Xuanchen Xiang and Simon Foo. *Recent Advances in Deep Reinforcement Learning Applications for Solving Partially Observable Markov Decision Processes (POMDP) Problems: Part I Fundamentals and Applications in Games, Robotics and Natural Language Processing*. *Machine Learning and Knowledge Extraction*, 3(3):554–581, 2021.
- [Xu 09] Zhe Xu, Travis Deyle, and Charles C Kemp. *1000 Trials: An empirically validated end effector that robustly grasps objects from the floor*. In 2009 IEEE International Conference on Robotics and Automation, pages 2160–2167. IEEE, 2009.
- [Xu 16] Zhe Xu and Emanuel Todorov. *Design of a highly biomimetic anthropomorphic robotic hand towards artificial limb regeneration*. In 2016 IEEE International Conference on Robotics and Automation (ICRA), pages 3485–3492. IEEE, 2016.
- [Yamaguchi 19] Akihiko Yamaguchi and Christopher G Atkeson. *Recent progress in tactile sensing and sensors for robotic manipulation: can we turn tactile sensing into vision?* *Advanced Robotics*, 33(14):661–673, 2019.
- [Yamakawa 11] Yuji Yamakawa, Akio Namiki, and Masatoshi Ishikawa. *Motion planning for dynamic folding of a cloth with two high-speed robot hands and two high-speed sliders*. In 2011 IEEE International Conference on Robotics and Automation, pages 5486–5491. IEEE, 2011.
- [Yamazaki 21] Kimitoshi Yamazaki and Taiki Abe. *A Versatile End-Effector for Pick-and-Release of Fabric Parts*. *IEEE Robotics and Automation Letters*, 6(2):1431–1438, 2021.
- [Yang 21] Zhihan Yang and Hai Nguyen. *Recurrent off-policy baselines for memory-based continuous control*. arXiv preprint arXiv:2110.12628, 2021.
- [Yarats 20] Denis Yarats, Ilya Kostrikov, and Rob Fergus. *Image augmentation is all you need: Regularizing deep reinforcement learning from pixels*. In International Conference on Learning Representations, 2020.
- [Yarats 21] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. *Mastering visual continuous control: Improved data-augmented reinforcement learning*. arXiv preprint arXiv:2107.09645, 2021.

- [Yoshikawa 90] Tsuneo Yoshikawa. *Foundations of robotics: analysis and control*. MIT press, 1990.
- [Yoshimi 12] Takashi Yoshimi, Naoyuki Iwata, Makoto Mizukawa, and Yoshinobu Ando. *Picking up operation of thin objects by robot arm with two-fingered parallel soft gripper*. In 2012 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), pages 7–12. IEEE, 2012.
- [Yuan 18] Wenzhen Yuan, Yuchen Mo, Shaoxiong Wang, and Edward H Adelson. *Active clothing material perception using tactile sensing and deep learning*. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 4842–4849. IEEE, 2018.
- [Yuan 20] Shenli Yuan, Lin Shao, Connor L Yako, Alex Gruebele, and J Kenneth Salisbury. *Design and control of roller grasper v2 for in-hand manipulation*. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 9151–9158. IEEE, 2020.
- [Yuba 17] Hiroyuki Yuba, Solvi Arnold, and Kimitoshi Yamazaki. *Unfolding of a rectangular cloth from unarranged starting shapes by a Dual-Armed robot with a mechanism for managing recognition error and uncertainty*. *Advanced Robotics*, 31(10):544–556, 2017.
- [Zamani 15] Bahareh Zamani, Magdalena Svanström, Gregory Peters, and Tomas Rydberg. *A carbon footprint of textile recycling: A case study in Sweden*. *Journal of industrial ecology*, 19(4):676–687, 2015.
- [Zhu 19] 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.
- [Zhu 20] Henry Zhu, Justin Yu, Abhishek Gupta, Dhruv Shah, Kristian Hartikainen, Avi Singh, Vikash Kumar, and Sergey Levine. *The ingredients of real-world robotic reinforcement learning*. arXiv preprint arXiv:2004.12570, 2020.
- [Zhu 22] Wangshu Zhu and Andre Rosendo. *PSTO: Learning Energy-Efficient Locomotion for Quadruped Robots*. *Machines*, 10(3):185, 2022.
- [Zwart 22] Sjoerd Zwart. *Engineering Epistemology: Between Theory and Practice*, 2022.

Appendix A

Human Observation Exercise

A.1 Consent Form

This Chapter contains the additional figures and details of the human observation exercise involving fabric manipulation referenced in Chapter 3. The first information presented in this appendix is the experiment information and consent form given to participants, shown across the following four pages.



**UNIVERSITY OF
CANBERRA**

Participant Information Form

Project Title

Humans and Cloth Manipulation: Deriving inspirations for robotic manipulation

Researcher

Mr. David Hinwood

PhD Candidate | Human-Centred Computing Laboratory | University of Canberra, Australia

Phone: +61 405 687 7772 | E-mail: David.Hinwood@canberra.edu.au

Supervisory Panel

Dr. Damith C. Herath

Associate Professor in Robotics | Human-Centred Computing Laboratory

Faculty of Education, Science, Technology and Mathematics. | University of Canberra, Australia

E-mail: Damith.Herath@canberra.edu.au

Supervisory Panel

Dr. Roland Goecke

Associate Professor in Affective Computing | Human-Centred Computing Laboratory

Faculty of Education, Science, Technology and Mathematics. | University of Canberra, Australia

E-mail: Roland.Goecke@canberra.edu.au

Project Aim

The aim of this exercise is to gain an understanding of how humans approach the problem of fabric handling.

Benefits of the Project

The information gained from the research will be used to design robot hands that manipulate fabric and gain a deeper understanding of human manipulation tendencies.

General Outline of the Project

The project will contribute towards a PhD thesis and possibly produce an academic publication(s) that develop robotic manipulators and techniques to handle fabric.

Participant Involvement

Participants who agree to participate in the research will be asked to:

1. Perform a series of manipulation tasks with a range of garments.
2. Be observed and recorded while performing the task.
3. Take part in a short questionnaire after performing the tasks.

Participation in the research is completely voluntary and participants may, without any penalty, decline to take part or withdraw at any time without providing an explanation or refuse to answer a question.

Confidentiality

Only the researcher/s will have access to the individual information provided by participants. Privacy and confidentiality will be assured at all times. The research outcomes may be presented at conferences and written up for publication. However, in all these publications, the privacy and confidentiality of individuals will be protected.

Anonymity

All reports and publications of the research will contain no information that can identify any individual and all information will be kept in the strictest confidence.

Data Storage

The information collected will be stored securely on a password protected computer throughout the project and then stored at the University of Canberra for the required five year period after which it will be destroyed according to university protocols.

Ethics Committee Clearance

The project has been approved by the Human Research Ethics Committee of the University of Canberra (6982 - Humans and Cloth Manipulation: Deriving inspirations for robotic manipulation).

Queries and Concerns

Queries or concerns regarding the research can be directed to the researcher and/or supervisor. Their contact details are at the top of this form. You can also contact the University of Canberra's Research Ethics & Integrity Unit at the address humanethicscommittee@canberra.edu.au.

If you would like some guidance on the questions you could ask about your participation please refer to the Participants' Guide located at <http://www.canberra.edu.au/ucresearch/attachments/pdf/am/Agreeing-to-participate-in-research.pdf>

Consent Form

Project Title

Humans and Cloth Manipulation: Deriving inspirations for robotic manipulation

Consent Statement

I have read and understood the information about the research. I am not aware of any condition that would prevent my participation, and I agree to participate in this project. I have had the opportunity to ask questions about my participation in the research. All questions I have asked have been answered to my satisfaction.

Please indicate whether you agree to participate in each of the following parts of the research (please indicate which parts you agree to by putting a cross in the relevant box):

- Performing a series of fabric manipulation tasks while being visually recorded and observed.
- Participate in a questionnaire once the manipulation tasks have been complete.

Name.....

Signature.....

Date

A summary of the research report can be forwarded to you when published. If you would like to receive a copy of the report, please include your mailing (or email) address below.

Name.....

Address.....

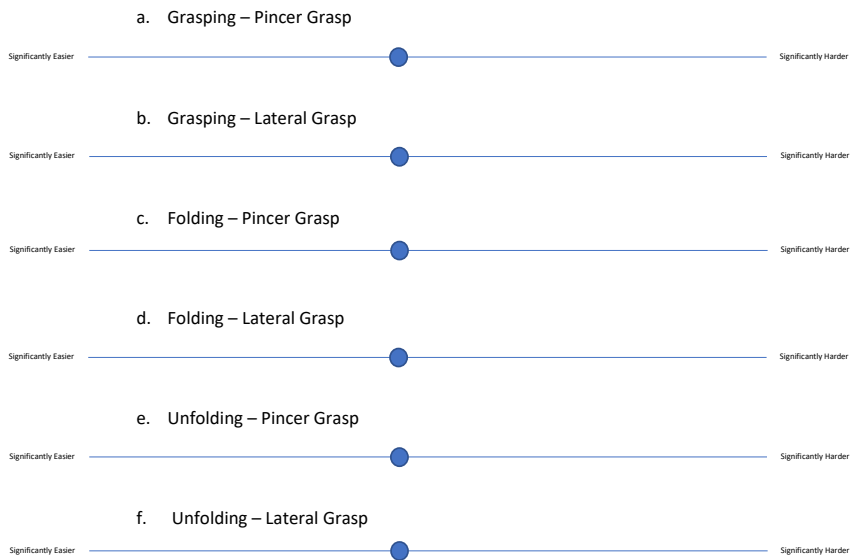
.....

A.2 Questionnaire

This section contains the attached questionnaire form that participants completed once completing the HRI exercises referenced in Chapter of 3. Each participant filled out this form upon completion of the observation exercise detailed in Section 3.5.

Observed Human Dexterous Manipulation: Questionnaire

1. Basic Details
 - a. Age
 - b. Gender Male | Female | Other | Prefer not to say
2. Did the difficulty of the tasks change over time when compared to the unconstrained condition (your first attempt), please circle the appropriate response to each action?
 - a. Grasping – Pincer Grasp Easier | Remained about the same | Harder
 - b. Grasping – Lateral Grasp Easier | Remained about the same | Harder
 - c. Folding – Pincer Grasp Easier | Remained about the same | Harder
 - d. Folding – Lateral Grasp Easier | Remained about the same | Harder
 - e. Unfolding – Pincer Grasp Easier | Remained about the same | Harder
 - f. Unfolding – Lateral Grasp Easier | Remained about the same | Harder
3. If there was a change in difficulty under the following conditions when compared to the unconstrained condition (your first attempt), by what degree, place a mark on the line where appropriate?
 - a. Grasping – Pincer Grasp



4. Were there any restrictions present that made certain actions difficult, if so, please provide details?

A.3 HRI Data and Information

The actual video feed of the human observation exercise is present at the University of Canberra’s Human-Centred Technology (HCT) lab. Please contact the appropriate faculty member or author for further details. This component of the appendix details the data collected throughout the observation exercise described in Section 3.5.

A.3.1 Analysed duration data in human observation exercise

Duration Data

Table A.1: The duration data (in seconds) of participants performing various tasks in the human observation exercise, highlighted blank yellow cells indicate errors made by a minority of participants.

Duration of tasks in seconds	Participants																			
Task Name	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10	P 11	P 12	P 13	P 14	P 15	P 16	P 17	P 18	P 19	P 20
Folding Shirt- Unconstrained	8	9	16	4	8	9	19	13	6	10	8	14	11	11	12	11	8	9	8	11
Folding Shirt- Unconstrained	9	7	13	4	7	8	29	11	8	9	6	13	9	11	9	9	4	8	11	9
Folding Shirt- Unconstrained	8	7	14	6	6	8		9	6	8	6	8	10	12	9	9	6	7	10	9
Folding Shirt- Unconstrained (Avg)	8.333	7.67	14.3	4.67	7	8.33		11	6.67	9	6.67	11.7	10	11.3	10	9.67	6	8	9.67	9.67
Sorting – Unconstrained	31	40	84	32	45	35	43	53	35	51	35	51	49	47	66	63	30	20	36	26
Folding/Sorting - Unconstrained	126	123	280	131	88	117	253	125	114	117	98	134	170	249	134	163	96	106	133	73
Folding Shirt 1 - Pinch	12	11	24	8	9	13	35	21	9	10	7	9	15	14	12	13	8	20	8	17
Folding Shirt 2 - Pinch	11	8	18	6	7		28	19	6	8	6	7	9	19	9	8	9	13	10	7
Folding Shirt 3 - Pinch	10	9	19	5	9		24	11	6	9	6	7	10	24	10	8	11	16	8	8
Folding Shirt- Pinch (Avg)	11	9.33	20.3	6.33	8.33		29	17	7	9	6.33	7.67	11.3	19	10.3	9.67	9.33	16.3	8.67	10.7
Sorting – Pinch	25	36	68	27	42	33	43	36	36	33		33	41	53	44	53	29	36	25	34
Folding/Sorting - Pinch	151	139	331	127	105	151	332	155	93	125	111	150	149	347	221	171	166	153	161	102
Folding Shirt 1 - Lateral	16	7	24	7	12	21	33	16	8	10	6	9	15	28	17	12	8	18	10	10
Folding Shirt 2 - Lateral	14	10	27	6	9	10	33	20	8	10	6	7	10	28	17	11	9	24	9	8
Folding Shirt 3 - Lateral	12	9	19	5	10	9		16	9	10	7	9	9	22	11	11	13	14	8	8
Folding Shirt- Lateral (Avg)	14	8.67	23.3	6	10.3	13.3		17.3	8.33	10	6.33	8.33	11.3	26	15	11.3	10	18.7	9	8.67
Sorting – Lateral	27	36	86	23	38	39	37	46	28	29	42	32	32	87	66	55	34	30	23	27
Folding/Sorting - Lateral	147	168	323	124	125	178	344	206	100	160	113	176	142	561	222	177	167	186	142	107

Table A.2: Averages of the participant’s times while folding a shirt.

ID	Unconstrained	Pincer	Lateral
1	8.333	11.000	14.000
2	7.667	9.333	8.667
3	14.333	20.333	23.333
4	4.667	6.333	6.000
5	7.000	8.333	10.333
6	11.000	17.000	17.333
7	6.667	7.000	8.333
8	9.000	9.000	10.000
9	11.667	7.667	8.333
10	10.000	11.333	11.333
11	11.333	19.000	26.000
12	10.000	10.333	15.000
13	9.667	9.667	11.333
14	6.000	9.333	10.000
15	8.000	16.333	18.667
16	9.667	8.667	9.000
17	9.667	10.667	8.667

Table A.3: Participant times while sorting clothing.

ID	Unconstrained	Pincer	Lateral
1	31	25	27
2	40	36	36
3	84	68	86
4	32	27	23
5	45	42	38
6	53	36	46
7	35	36	28
8	51	33	29
9	51	33	32
10	49	41	32
11	47	53	87
12	66	44	66
13	63	53	55
14	30	29	34
15	20	36	30
16	36	25	23
17	26	34	27

Table A.4: Participant times while folding and sorting clothing.

ID	Unconstrained	Pincer	Lateral
1	126	151	147
2	123	139	168
3	280	331	323
4	131	127	124
5	88	105	125
6	125	155	206
7	114	93	100
8	117	125	160
9	134	150	176
10	170	149	142
11	249	347	561
12	134	221	222
13	163	171	177
14	96	166	167
15	106	153	186
16	133	161	142
17	73	102	107

Participant questionnaire responses

Table A.5: The recorded responses of the degree of difficulty participants found the pincer or lateral constraints compared to the unconstrained. This image is from the Jamovi software.

ID	Pincer_folding (%)	Lateral_folding (%)	Pincer_unfolding (%)	Lateral_unfolding (%)	Pincer_grasping (%)	Lateral_grasping (%)
1	40	60	20	20	20	40
2	40	60	0	0	20	30
3	0	30	0	30	0	30
4	0	80	0	20	0	40
5	30	100	20	40	10	60
6	35	50	0	0	0	38
7	20	40	40	40	10	10
8	0	75	0	85	0	80
9	60	100	0	0	0	20
10	-20	40	-60	60	40	-80
11	20	80	20	60	0	20
12	60	80	0	0	0	0
13	80	60	60	80	20	60
14	18	38	5	20	9	55
15	0	60	0	0	0	60
16	60	20	-80	20	0	20
17	25	60	10	8	10	22
18	0	60	0	20	0	20
19	20	40	0	20	0	40
20	40	80	40	80	20	40

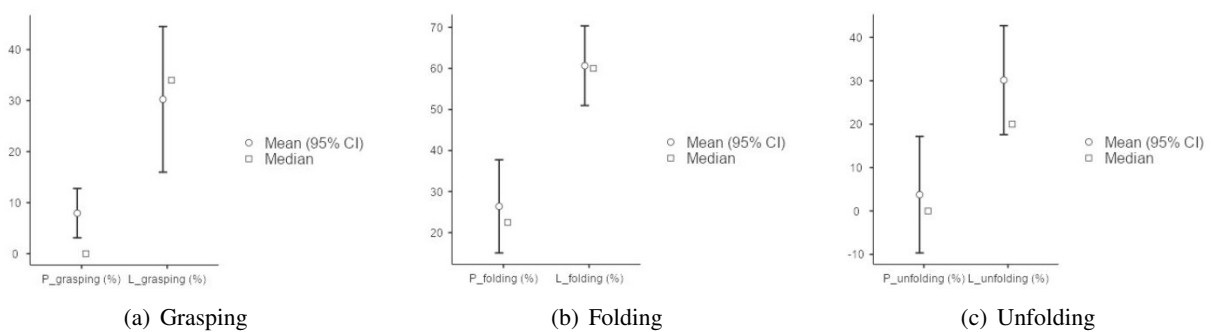


Figure A.1: A visualisation from the ANOVA analysis of Jamovi showing the degree of difficulty participants found pincer and lateral conditions compared to the unconstrained setting while grasping, folding and unfolding. The y-axis represents the difficulty magnitude responses on the form in Appendix A.2.

Appendix B

Mathematical Modelling of the Effector

B.0.1 Homogeneous Transformations

The homogeneous transformation matrix is a concise method in linear algebra of mapping displacement in both position and rotation of coordinate frames commonly used in robotic modelling. When representing transformations in three-dimensional space, the matrix is a size of 4×4 which comprehensively describes the rotation and displacement from an arbitrary frame d to a target frame t . For example, this transformation can alternatively be present in the form of Equation B.1, where dR_t represents the rotation matrix transformation from frame d to frame t and dp_t represents the position displacement vector.

$${}^dH_t = \begin{bmatrix} {}^dR_t & {}^dp_t \\ 0 & 1 \end{bmatrix} \quad (\text{B.1})$$

The complete form of the Homogeneous transformation matrix is present in Equation B.2, highlighting the rotation matrix R and the position vector p .

$$H = \begin{bmatrix} R_{11} & R_{12} & R_{13} & p_x \\ R_{21} & R_{22} & R_{23} & p_y \\ R_{31} & R_{32} & R_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.2})$$

The properties of homogeneous transformations include that they can be chained together via matrix multiplication resulting in a single matrix representing a series of transformations. For example, equation B.3 displays this operation where from frame w , the transformation matrix to frame t concerning frame d .

$${}^w H_t = {}^w H_d \cdot {}^d H_t \quad (\text{B.3})$$

Throughout this thesis, it becomes necessary to express the combination of rotation and translation components independently, the notation of $T_x(a)$ (Representing a linear transform of distance a along the x-axis) and $R_x(r)$ (Representing a rotation of r radians around the x-axis) represent transformations in robotic space. These coordinate transforms are chained together as $T_x(a)R_x(r)$, representing sequential translation and rotation actions.

B.0.2 Properties of the Rotation Matrix

The basics: While representing rotation in three-dimensional space, the rotation matrix is of a 3×3 size as shown previously in Equation B.2. However, the rotation matrix can also exist in the following form shown in Equation B.4.

$$R = [x' y' z'] = \begin{bmatrix} x'_x & y'_x & z'_x \\ x'_y & y'_y & z'_y \\ x'_z & y'_z & z'_z \end{bmatrix} \quad (\text{B.4})$$

where three unit vectors of x', y', z' describe the orientation of a frame concerning a reference frame. The notation in Equation B.4 denotes the direction cosines from frame $O' - x'y'z'$ to a base reference frame $O - xyz$, expressed in the subscript component. With these definitions, Equations B.5 - B.7 express the rotation matrix moving a body around the available axis in 3D space.

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix} \quad (\text{B.5})$$

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad (\text{B.6})$$

$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{B.7})$$

Rotation matrices can also multiply together, resulting in sequential rotations analogous to the described sequential transformations described in Section B.0.1.

Deriving the roll, pitch and yaw values: It is sometimes more intuitive to represent the rotation matrix as a vector representing three sequential rotations rather than the standard 3×3 matrix form. This representation is particularly applicable when multiple rotation matrices have sequentially multiplied together. Such a method derives the roll, pitch and yaw values or the Euler angles. In this thesis, part of the operational inverse kinematics solution depends on deriving the roll, pitch and yaw angles. The first thing is to consider in what order the roll pitch and yaw values can be derived.

In this instance, the method assumes a rotation order of z , y and x . Suppose one considers that a rotation matrix as a sequence of 3 rotations as matrix R , from Equation B.4, where $R = R_z(\phi)R_y(\theta)R_x(\psi)$. Then, the roll pitch and yaw values are calculable via Equations B.8 - B.10. However, previous authors note limits between rotational ranges of $-\frac{pi}{2}$ to $\frac{pi}{2}$ while using these equations. Further details are present in textbooks. See [Sciavicco 12].

$$\phi = \text{Atan2}(R_{21}, R_{11}) \quad (\text{B.8})$$

$$\theta = \text{Atan2}\left(-R_{31}, \sqrt{R_{32}^2 + R_{33}^2}\right) \quad (\text{B.9})$$

$$\psi = \text{Atan2}(R_{32}, R_{33}) \quad (\text{B.10})$$

Deriving a rotation matrix: As rotation matrices represent a frame's orientation, this expression's derivative represents the angular velocity. The inherent properties of the rotation matrix can infer an expression representing the derivative. Rotation matrices are orthogonal, resulting in the expression shown in Equation B.11 where i denotes an arbitrary axis of rotation, I_3 being an identity matrix and θ represents the corresponding displacement. With this property and the chain rule, we can write the expression of the derivative of this matrix as shown in Equation B.12. The resulting term $\frac{d}{d\theta}R_iR_i^T$ is a skew-symmetric matrix. Thus, one can express the matrix in a form as shown in Equations B.14 and B.13.

$$R_i(\theta)R_i(\theta)^T = I_3 \quad (\text{B.11})$$

$$\frac{d}{d\theta} R_i R_i^T + \left(\frac{d}{d\theta} R_i R_i^T \right)^T = 0 \quad (\text{B.12})$$

$$S = \frac{d}{d\theta} R_i R_i^T \quad (\text{B.13})$$

$$S + S^T = 0 \quad (\text{B.14})$$

The nature of a skew-symmetric matrix has terms reflected across the diagonal as shown in Equation B.15. One can also represent the skew-symmetric matrix with the vector shown in Equation B.16.

$$S(v) = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \quad (\text{B.15})$$

$$v = [x, y, z] \quad (\text{B.16})$$

With these definitions, one can succinctly express the derivative of a rotation matrix around the x, y and z axis with Equations B.17 - B.19, respectively.

$$\frac{d}{d\theta} R_x = R_x \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (\text{B.17})$$

$$\frac{d}{d\theta} R_y = R_y \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (\text{B.18})$$

$$\frac{d}{d\theta} R_z = R_z \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (\text{B.19})$$

Finally, we can then express the total angular velocity of a reference frame when a velocity is applied on axis i at a speed of $\dot{\theta}$ via the expression shown in Equation B.20.

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = R_i v^T \dot{\theta} \quad (\text{B.20})$$

B.0.3 Gripper Kinematics

Kinematics and Kinostatics calculation via symbolic toolbox

This script calculates the position, velocity and forces relationships of the gripper. Note that this script requires the robotics toolbox by Peter Corke to be executed.

The symbolic kinematic variables from the system are first created and implemented.

```
close all;clear all;
%Symbolic Variables of the kinematic system
syms d_1 a_1 a_2 a_3 q_0 q_1 q_2 q_3 q_dot_0 q_dot_1 ...
      q_dot_2 q_dot_3 Delta_z;
sym_pi = sym(pi);
%Velocity Vector
q_dot = [q_dot_0; q_dot_1; q_dot_2; q_dot_3];

%Position Vector
q = [q_0; q_1; q_2; q_3];

%Base transform of the DH parameters
FK_sequence = trotz(sym_pi/2) * trotx(sym_pi/2) * trotz(sym_pi) ...
             * transl(0,0,q_0) * trotx(sym_pi/2) * trotz(q_1) * ...
             transl(a_1,0,d_1) * trotx(sym_pi/2) * trotz(q_2) * ...
             transl(a_2,0,0) * trotx(0) * trotz(q_3) *...
             transl(a_3,0,0) * trotx(-sym_pi/2) ;

%Base transform including
Sensor_FK_sequence = FK_sequence * transl([0 0 Delta_z])* troty(sym_pi/4);
```

This section presents the FK kinematic equations for both the TCP of the DH parameters and the sensor position.

```
%establish the standard FK equations
x_fk = simplify(FK_sequence(1,4))
```

$$x_fk = q_0 - a_3 (\sin(q_1) \sin(q_2) \sin(q_3) - \cos(q_2) \cos(q_3) \sin(q_1)) + a_1 \sin(q_1) + a_2 \cos(q_2) \sin(q_1)$$

```
y_fk = simplify(FK_sequence(2,4))
```

$$y_fk = -\cos(q_1) (a_1 + a_3 \cos(q_2 + q_3) + a_2 \cos(q_2))$$

```
z_fk = simplify(FK_sequence(3,4))
```

$$z_fk = d_1 + a_3 \sin(q_2 + q_3) + a_2 \sin(q_2)$$

```
p_fk = -q_2-q_3
```

$$p_fk = -q_2 - q_3$$

```
%Converting the symbolic expressions to functions accepting numerical
%values
%x_fk = q_0 - a_3*(sin(q_1)*sin(q_2)*sin(q_3) - cos(q_2)*
% cos(q_3)*sin(q_1)) + a_1*sin(q_1) + a_2*cos(q_2)*sin(q_1)
```

```

syms x_FK_calc(q_0,q_1,q_2,q_3,a_1,a_2,a_3)
x_FK_calc(q_0,q_1,q_2,q_3,a_1,a_2,a_3) = x_fk;

%y_fk = -cos(q_1)*(a_1 + a_3*cos(q_2 + q_3) + a_2*cos(q_2))
syms y_FK_calc(q_1,q_2,q_3,a_1,a_2,a_3)
y_FK_calc(q_1,q_2,q_3,a_1,a_2,a_3) = y_fk;

%z_fk = d_1 + a_3*sin(q_2 + q_3) + a_2*sin(q_2)
syms z_FK_calc(q_2,q_3,d_1,a_2,a_3)
z_FK_calc(q_2,q_3,d_1,a_2,a_3) = z_fk;

%p_fk = -q_2-q_3
syms p_FK_calc(q_2,q_3)
p_FK_calc(q_2,q_3) = p_fk;

```

The forward kinematic equations below define the position of the sensor from

the transforms of $T_z(\Delta_z)R_x\left(\frac{\pi}{4}\right)$ applied to the traditional DH parameters.

```

%establish the sensor modified FK equations
x_s_fk = simplify(Sensor_FK_sequence(1,4))

```

$$x_s_fk = q_0 - a_3 (\sin(q_1) \sin(q_2) \sin(q_3) - \cos(q_2) \cos(q_3) \sin(q_1)) + a_1 \sin(q_1) - \Delta_z (\cos(q_2) \sin(q_1) \sin(q_3) + \cos(q_3))$$

```

y_s_fk = simplify(Sensor_FK_sequence(2,4))

```

$$y_s_fk = -\cos(q_1) (a_1 - \Delta_z \sin(q_2 + q_3) + a_3 \cos(q_2 + q_3) + a_2 \cos(q_2))$$

```

z_s_fk = simplify(Sensor_FK_sequence(3,4))

```

$$z_s_fk = d_1 + \Delta_z \cos(q_2 + q_3) + a_3 \sin(q_2 + q_3) + a_2 \sin(q_2)$$

```

p_s_fk = -q_2-q_3 + sym_pi/4

```

```

p_s_fk =

```

$$\frac{\pi}{4} - q_3 - q_2$$

```

%Converting the symbolic expressions to functions accepting numerical
%values
%x_s_fk = q_0 - a_3*(sin(q_1)*sin(q_2)*sin(q_3) - cos(q_2)
% *cos(q_3)*sin(q_1))
% - Delta_z*(cos(q_2)*sin(q_1)*sin(q_3) + cos(q_3)*sin(q_1)*sin(q_2)) +
% a_1*sin(q_1) + a_2*cos(q_2)*sin(q_1)
syms x_s_fk_calc(q_0,q_1,q_2,q_3,a_1,a_2,a_3, Delta_z);
x_s_fk_calc(q_0,q_1,q_2,q_3,a_1,a_2,a_3, Delta_z) = x_s_fk;

%y_s_fk = -cos(q_1)*(a_1 + a_3*cos(q_2 + q_3) -
% Delta_z*sin(q_2 + q_3) + a_2*cos(q_2))

```

```

syms y_s_fk_calc(q_1,q_2,q_3,a_1,a_2,a_3, Delta_z);
y_s_fk_calc(q_1,q_2,q_3,a_1,a_2,a_3, Delta_z) = y_s_fk;

%z_s_fk = d_1 + Delta_z*cos(q_2 + q_3) + a_3*sin(q_2 + q_3)
% + a_2*sin(q_2)
syms z_s_fk_calc(q_2,q_3,d_1,a_2,a_3, Delta_z);
z_s_fk_calc(q_2,q_3,d_1,a_2,a_3, Delta_z) = z_s_fk;

%p_s_fk = pi/4 - q_3 - q_2
syms p_s_fk_calc(q_2,q_3,pi);
p_s_fk_calc(q_2,q_3,pi) = p_s_fk;

```

```
%The inverse series of calculations
```

```
syms X Y Z P
```

```
q0_ik = X - sin(q_1)*(a_1 + a_3*cos(q_2 + q_3)+a_2*cos(q_2))
```

$$q0_{ik} = X - \sin(q_1) (a_1 + a_3 \cos(q_2 + q_3) + a_2 \cos(q_2))$$

```
q1_ik = sym_pi - acos(Y/(a_1 + a_3*cos(q_2 + q_3)+a_2*cos(q_2)))
```

```
q1_ik =
```

$$\pi - \arccos\left(\frac{Y}{a_1 + a_3 \cos(q_2 + q_3) + a_2 \cos(q_2)}\right)$$

```
q2_ik = asin((Z-d_1+a_3*sin(P))/a_2)
```

```
q2_ik =
```

$$\arcsin\left(\frac{Z - d_1 + a_3 \sin(P)}{a_2}\right)$$

```
q3_ik = -P -q_2
```

$$q3_{ik} = -P - q_2$$

```

syms q0_ik_calc(q_1,q_2,q_3,a_1,a_2,a_3,X);
q0_ik_calc(q_1,q_2,q_3,a_1,a_2,a_3,X) = q0_ik;

syms q1_ik_calc(q_2,q_3,a_1,a_2,a_3,Y,sym_pi);
q1_ik_calc(q_2,q_3,a_1,a_2,a_3,Y,sym_pi) = q1_ik;

syms q2_ik_calc(d_1,a_2,a_3,Z,P);
q2_ik_calc(d_1,a_2,a_3,Z,P) = q2_ik;

syms q3_ik_calc(q_2,P);
q3_ik_calc(q_2,P) = q3_ik;

```

This section of code parses through data from various joint positions to validate that the forward and inverse kinematic algorithms are correct across the manipulator workspace, may be modified to display data outputs.

```

%if requiring comparisons to print, enable
debug_flag = false;
%build the joint limits
%establish the joint limits from the DH parameters
nS = 4;
pi_doub = 3.14159265;

q_0_joint_poses = linspace(0,0.079,nS);

q_1_joint_poses = linspace(0.8727,2.2689,nS);
q_1_joint_poses(nS+1) = 0;

q_2_joint_poses = linspace(0,pi_doub/2,nS);

q_3_joint_poses = linspace(-1.7453,0.0175,nS);
q_3_joint_poses(nS+1) = 0;
q_3_joint_poses(nS+2) = -pi_doub/2;

a1 = 19.958/1000;
a2 = 35.752/1000;
a3 = 44.491/1000;
d1 = 18.582/1000;
Delta_z = -13.363/1000;
q3_alt_offset = atan(Delta_z/a3);
a3_alt = sqrt(Delta_z^2 + a3^2);

rounder_estimator = 3;

for qI0 = 1 : length(q_0_joint_poses)
    for qI1 = 1 : length(q_1_joint_poses)
        for qI2 = 1 : length(q_2_joint_poses)

            for qI3 = 1 : length(q_3_joint_poses)

                q0 = q_0_joint_poses(qI0);
                q1 = q_1_joint_poses(qI1);
                q2 = q_2_joint_poses(qI2);
                q3 = q_3_joint_poses(qI3);

                bFK_x = double(x_FK_calc(q0,q1,q2,q3,a1,a2,a3));
                bFK_y = double(y_FK_calc(q1,q2,q3,a1,a2,a3));
                bFK_z = double(z_FK_calc(q2,q3,d1,a2,a3));
                bFK_p = double(p_FK_calc(q2,q3));
            end
        end
    end
end

```

```

sFK_x = double(x_s_fk_calc(q0,q1,q2,q3,a1,a2,a3,Delta_z));
sFK_y = double(y_s_fk_calc(q1,q2,q3,a1,a2,a3,Delta_z));
sFK_z = double(z_s_fk_calc(q2,q3,d1,a2,a3,Delta_z));
sFK_p = double(p_s_fk_calc(q2,q3,pi_doub));

s_bFK_x = double(x_FK_calc(q0,q1,q2,q3+q3_alt_offset,a1,a2,a3_alt));
s_bFK_y = double(y_FK_calc(q1,q2,q3+q3_alt_offset,a1,a2,a3_alt));
s_bFK_z = double(z_FK_calc(q2,q3+q3_alt_offset,d1,a2,a3_alt));
%The offset only is used in position calculations
s_bFK_p = double(p_FK_calc(q2,q3))+(pi_doub/4);

compSens = [sFK_x,sFK_y,sFK_z,sFK_p];
compFK_solution = [s_bFK_x,s_bFK_y,s_bFK_z,s_bFK_p];

comparison = ...
    all(round(abs(compSens-compFK_solution),rounder_estimator)...
        < 1e4*eps(min(abs(compSens),abs(compFK_solution))));

if debug_flag
    disp('Calc comparison')
    disp(compSens)
    disp(compFK_solution)
end
%if not equal, break
if ~(logical(comparison))
    disp(compSens);
    disp(compFK_solution);
    causeException = MException('MATLAB:valueError',...
        'The kinematic equations do not match. ');
    throw(causeException);
end

ik_result_q2 = double(q2_ik_calc(d1,a2,a3,bFK_z,bFK_p));
ik_result_q3 = double(q3_ik_calc(ik_result_q2,bFK_p));
ik_result_q1 = double(q1_ik_calc(ik_result_q2,ik_result_q3,...
    a1,a2,a3,bFK_y,sym_pi));
ik_result_q0 = double(q0_ik_calc(ik_result_q1,ik_result_q2,...
    ik_result_q3,a1,a2,a3,bFK_x));

basicResultIK = [ik_result_q0,ik_result_q1,ik_result_q2,ik_result_q3];
originalJointPos = [q0,q1,q2,q3];

comparisonIK = ...
    all(round(abs(basicResultIK-originalJointPos),rounder_estimator) ...
        < 1e4*eps(min(abs(basicResultIK),abs(originalJointPos))));
if debug_flag
    disp('Calc comparison')
    disp(basicResultIK)

```

```

        disp(originalJointPos)
    end
    %if not equal or close, break
    if ~(logical(comparisonIK))
        disp(basicResultIK);
        disp(originalJointPos);
        causeException = MException('MATLAB:valueError',...
            'The kinematic equations do not match. ');
        throw(causeException);
    end

    %equations need to include the q3 offset
    ikSens_result_q2 = ...
        double(q2_ik_calc(d1,a2,a3_alt,s_bFK_z,...
            s_bFK_p-(pi_doub/4)-q3_alt_offset));
    ikSens_result_q3 = ...
        double(q3_ik_calc(ikSens_result_q2,s_bFK_p-(pi_doub/4)));
    ikSens_result_q1 = ...
        double(q1_ik_calc(ikSens_result_q2,...
            ikSens_result_q3+q3_alt_offset,...
            a1,a2,a3_alt,s_bFK_y,sym_pi));
    ikSens_result_q0 = ...
        double(q0_ik_calc(ikSens_result_q1,ikSens_result_q2,...
            ikSens_result_q3+q3_alt_offset,a1,a2,a3_alt,s_bFK_x));

    basicSensResultIK = ...
        [ikSens_result_q0,ikSens_result_q1,ikSens_result_q2,...
            ikSens_result_q3];
    originalJointPos = [q0,q1,q2,q3];

    comparisonSIK = ...
        all(round(abs(basicSensResultIK-originalJointPos),rounder_estimator) ...
            < 1e4*eps(min(abs(basicSensResultIK),abs(originalJointPos))));
    %if not equal, break
    if debug_flag
        disp('Calc comparison')
        disp(basicSensResultIK)
        disp(originalJointPos)
    end
    if ~(logical(comparisonIK))
        disp(basicResultIK);
        disp(originalJointPos);
        causeException = MException('MATLAB:valueError',...
            'The sensing kinematic equations do not match. ');
        throw(causeException);
    end
end
end
end
end

```

```
end
disp('calculations correct');
```

calculations correct

The next section of code uses the forward kinematic equations to model the jacobian matrix for both velocity and force calculations throughout the system.

```
%establish the jacobian
dhJacobian = [diff(x_fk,q_0),diff(x_fk,q_1),diff(x_fk,q_2),diff(x_fk,q_3);
              diff(y_fk,q_0),diff(y_fk,q_1),diff(y_fk,q_2),diff(y_fk,q_3);
              diff(z_fk,q_0),diff(z_fk,q_1),diff(z_fk,q_2),diff(z_fk,q_3);
              diff(p_fk,q_0),diff(p_fk,q_1),diff(p_fk,q_2),diff(p_fk,q_3)];

inverseDHJacobian = inv(dhJacobian);
transposeDHJacobian = transpose(dhJacobian);
determinant = simplify(det(dhJacobian))
```

$$\text{determinant} = -a_2 \cos(q_2) \sin(q_1) (a_1 + a_3 \cos(q_2 + q_3) + a_2 \cos(q_2))$$

This section establishes the Jacobian matrix for the sensor, also known as $J_s(q)$.

```
%establish the sensor jacobian
sensorJacobian = [diff(x_s_fk,q_0),diff(x_s_fk,q_1),diff(x_s_fk,q_2),diff(x_s_fk,q_3);
                  diff(y_s_fk,q_0),diff(y_s_fk,q_1),diff(y_s_fk,q_2),diff(y_s_fk,q_3);
                  diff(z_s_fk,q_0),diff(z_s_fk,q_1),diff(z_s_fk,q_2),diff(z_s_fk,q_3);
                  diff(p_s_fk,q_0),diff(p_s_fk,q_1),diff(p_s_fk,q_2),diff(p_s_fk,q_3)];

inversesensorJacobian = inv(sensorJacobian);
transposesensorJacobian = transpose(sensorJacobian);
determinant_sensor = simplify(det(sensorJacobian))
```

$$\text{determinant_sensor} = -a_2 \cos(q_2) \sin(q_1) (a_1 - \Delta_z \sin(q_2 + q_3) + a_3 \cos(q_2 + q_3) + a_2 \cos(q_2))$$

Create a latex version of the jacobian formatted for thesis.

```
%format a latex jacobian for thesis document
dhJacobianLatex = latex(simplify(dhJacobian));
dhJacobianLatex = strrep(dhJacobianLatex, '\cos', 'c');
dhJacobianLatex = strrep(dhJacobianLatex, 'q_{2}+q_{3}', 'q_{2+3}');
dhJacobianLatex = strrep(dhJacobianLatex, '\sin', 's');
```


B.0.4 Dynamics Calculation and Validation

Langrangian formulation and validation of novel gripper

This script calculates the symbolic lagrangian, creates a model from the robotic toolbox and validates that the calculations are correct. The coriolis matrix, joint-space inertia matrix and gravity term variables are also calculated. Note that this script requires the robotics toolbox by Peter Corke to be executed.

The symbolic kinematic variables from the system are first created.

```
close all;clear all;
%Symbolic Variables of Our kinematic system
syms d_1 a_1 a_2 a_3 q_0 q_1 q_2 q_3 q_dot_0 q_dot_1 ...
      q_dot_2 q_dot_3;
sym_pi = sym(pi);
%Velocity Vector
q_dot = [q_dot_0; q_dot_1; q_dot_2; q_dot_3];

%Position Vector
q = [q_0; q_1; q_2; q_3];

%The gravity Vector
syms g_x g_y g_z;
g = [g_x;g_y;g_z];

%Base transform of the DH parameters
BaseTransform = trotx(sym_pi/2) * trotx(sym_pi/2);
%dictates the position of link 0 from the point of actuation as
% it remains a prismatic actuator
T_0j = BaseTransform * trotx(sym_pi) * transl(0,0,q_0);
R_0j = T_0j(1:3,1:3);

%Rotation Transforms to each link before each rotation action
%Row 0 of the DH parameters
T_1 = BaseTransform * trotx(sym_pi) * transl(0,0,q_0) * ...
      trotx(sym_pi/2) ;
R_1 = T_1(1:3,1:3);
%dictates the orientation of link 1 from the point of actuation
T_1j = T_1 * trotx(q_1);
R_1j = T_1j(1:3,1:3);

%Row 1 of the DH parameters
T_2 = T_1 * trotx(q_1) * transl(a_1,0,d_1) * trotx(sym_pi/2);
R_2 = T_2(1:3,1:3);
%dictates the orientation of link 2 from the point of actuation
T_2j = T_2 * trotx(q_2);
R_2j = T_2j(1:3,1:3);

%Row 2 of the DH parameters
T_3 = T_2 * trotx(q_2) * transl(a_2,0,0) * trotx(0) ;
R_3 = T_3(1:3,1:3);
%dictates the orientation of link 3 from the point of actuation
T_3j = T_3 * trotx(q_3);
```

```
R_3j = T_3j(1:3,1:3);

%Row 3 of the DH parameters, R indicates the rotation matrix
% only component
T_E = T_3 * trotx(q_3) * transl(a_3,0,0) * trotx(-sym_pi/2) ;
R_E = T_E(1:3,1:3);
```

The lagrangian function can be surmised as the potential energy of a system subtracted from the kinetic energy, i.e. $L = T - U$.

It can also take the form below where τ_i represents the force/torque needed from joint i for a desired position of a joint (q), velocity (\dot{q}) and acceleration (\ddot{q}). This problem can also be expressed as $q, \dot{q}, \ddot{q} \rightarrow \tau$, also known as the inverse dynamics problem.

$$\tau_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i}$$

However, this alternative form will become more applicable as described by the textbook "Foundations of Robotics, analysis and control by Yoshikawa".

$$\tau_i = \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i}$$

The dynamics modelling method starts by estimating the kinetic energy of each link.

The CoM (Centre of Mass) and mass of the links variables are now created. Each CoM variable is defined as the expression of the relative transform from the row of the denavit hartenberg parameters. That is, the transform for link i will be defined as the transform from the end of i th row of the DH parameters.

```
%Symbolic variables for unit masses of each link
syms m_l0 m_l1 m_l2 m_l3;

%We next dictate the COM equations, for prismatic links, we
% take the descriptor from the point of the next axis
syms p_l0x p_l0y p_l0z;
Vecp_l0 = [p_l0x; p_l0y; p_l0z];

%For rotational links, the descriptor is taken from
% the rotation point of axis
%COM link 1
syms p_l1x p_l1y p_l1z;
Vecp_l1 = [p_l1x; p_l1y; p_l1z];
%COM link 2
syms p_l2x p_l2y p_l2z;
Vecp_l2 = [p_l2x; p_l2y; p_l2z];
%COM link 3
syms p_l3x p_l3y p_l3z;
Vecp_l3 = [p_l3x; p_l3y; p_l3z];

%Per the convention of the robotics toolbox, each COM is
```

```

% determined from the transform of that particular row
% of the parameters COM link 0, These variables here represent the COM of
% each link relative to the base frame or 'W'
p_l0_model = Vecp_l0;
p_l0 = T_1 * transl(p_l0_model);
%COM link 1
p_l1_model = Vecp_l1;
p_l1 = T_2 * transl(p_l1_model);
%COM link 2
p_l2_model = Vecp_l2;
p_l2 = T_3 * transl(p_l2_model);
%COM link 3
p_l3_model = Vecp_l3;
p_l3 = T_E * transl(p_l3_model);

```

Finding the kinetic energy of a link.

The kinetic energy of a link can be found via two contributions, a translation and rotation component. These components are calculated separately below.

Translational Link Energy Calculation

The translational component T_T represents the translational component of the effector with m_i being the mass of the effector, and \dot{P}_i being the linear velocity of the center of mass.

$$T_T = \sum_{i=0}^3 \frac{1}{2} m_i \dot{P}_i^T \dot{P}_i$$

Where m_i is simply the mass of the link, \dot{p}_i is the derivative value of the equation that represents the COM position as a vector we first start by denoting P_i as the position vector of the COM for link i .

$$P_i = \begin{bmatrix} x_{p_i} \\ y_{p_i} \\ z_{p_i} \end{bmatrix}$$

Each element of P_i represents the COM position in the x, y and z axis with respect to the base coordinate frame 'W'. This section builds a jacobian for each COM link and multiplies it by the speed of the actuators resulting in the linear velocity for each center of mass.

This results in the linear velocity vector shown for each link;

$$\dot{P}_i = \begin{bmatrix} \dot{x}_{p_i} \\ \dot{y}_{p_i} \\ \dot{z}_{p_i} \end{bmatrix}$$

$$\dot{P}_i = \begin{bmatrix} \frac{\partial P_{ix}}{\partial q_0} & \dots & \frac{\partial P_{ix}}{\partial q_i} & 0 & \dots \\ \frac{\partial P_{iy}}{\partial q_0} & \dots & \frac{\partial P_{iy}}{\partial q_i} & 0 & \dots \\ \frac{\partial P_{iz}}{\partial q_0} & \dots & \frac{\partial P_{iz}}{\partial q_i} & 0 & \dots \end{bmatrix} \dot{q}$$

```

%The jacobian multiplied by the velocity vector to output the
% linear, COM position as a function of the whole system is
% also defined before each jacobian object velocities
[p_l0Rx, p_l0Ry, p_l0Rz] = transl(p_l0);
p_dot_l0 = [ diff(p_l0Rx,q_0),0,0,0;
            diff(p_l0Ry,q_0),0,0,0;
            diff(p_l0Rz,q_0),0,0,0]*q_dot;

[p_l1Rx, p_l1Ry, p_l1Rz] = transl(p_l1);
p_dot_l1 = [diff(p_l1Rx,q_0),diff(p_l1Rx,q_1),0,0;
            diff(p_l1Ry,q_0),diff(p_l1Ry,q_1),0,0;
            diff(p_l1Rz,q_0),diff(p_l1Rz,q_1),0,0]*q_dot;

[p_l2Rx, p_l2Ry, p_l2Rz] = transl(p_l2);
p_dot_l2 = [diff(p_l2Rx,q_0),diff(p_l2Rx,q_1),diff(p_l2Rx,q_2),0;
            diff(p_l2Ry,q_0),diff(p_l2Ry,q_1),diff(p_l2Ry,q_2),0;
            diff(p_l2Rz,q_0),diff(p_l2Rz,q_1),diff(p_l2Rz,q_2),...
            0]*q_dot;

[p_l3Rx, p_l3Ry, p_l3Rz] = transl(p_l3);
p_dot_l3 = [diff(p_l3Rx,q_0),diff(p_l3Rx,q_1),...
            diff(p_l3Rx,q_2),diff(p_l3Rx,q_3);
            diff(p_l3Ry,q_0),diff(p_l3Ry,q_1),...
            diff(p_l3Ry,q_2),diff(p_l3Ry,q_3);
            diff(p_l3Rz,q_0),diff(p_l3Rz,q_1),...
            diff(p_l3Rz,q_2),diff(p_l3Rz,q_3)]...
            *q_dot;

T_l0L = 1/2 * m_l0 * transpose(p_dot_l0)*p_dot_l0;
T_l1L = 1/2 * m_l1 * transpose(p_dot_l1)*p_dot_l1;
T_l2L = 1/2 * m_l2 * transpose(p_dot_l2)*p_dot_l2;
T_l3L = 1/2 * m_l3 * transpose(p_dot_l3)*p_dot_l3;

kineticTranslationalEnergy = T_l0L + T_l1L + T_l2L + T_l3L;

```

Rotation Link Energy Calculation

T_R represents the rotational component with the expression below.

$$T_R = \sum_{i=0}^3 \frac{1}{2} \omega_i^T I_i \omega_i$$

For each link, ω_i represents the angular velocity and I_i represents the inertia tensor about the COM. The inertia tensor is estimated through CAD software.

The inertia tensor/takes the form of a 3x3 matrix as shown below;

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}$$

To estimate the angular velocity of a link ω_i , we simply utilised a the rotational jacobian calculation from a geometric matrix. This is achieved by taking the derivative about the z axis of the rotation before the actuation. Thus the final rotational calculation is shown in the code segment below.

```
%Symbolic variables for unit inertia tensors about each link,
% as we do not dictate the torque of the stepper m, such a
% request becomes an unnessasry descriptor
syms I_11xx I_11yy I_11zz I_11xy I_11yz I_11xz;
syms I_12xx I_12yy I_12zz I_12xy I_12yz I_12xz;
syms I_13xx I_13yy I_13zz I_13xy I_13yz I_13xz;

%As I_10 relates to a prismatic joint, the inertia tensor is
% given values of zero since no rotational motion takes place
I_10 = [0,0,0;0,0,0;0,0,0];
I_11 = [I_11xx,I_11xy,I_11xz;I_11xy,I_11yy,I_11yz;...
        I_11xz,I_11yz,I_11zz];
I_12 = [I_12xx,I_12xy,I_12xz;I_12xy,I_12yy,I_12yz;...
        I_12xz,I_12yz,I_12zz];
I_13 = [I_13xx,I_13xy,I_13xz;I_13xy,I_13yy,I_13yz;...
        I_13xz,I_13yz,I_13zz];

%The rotational component of our systems geometric jacobian,
% this takes the derivative of rotation matrix components
% from the DH parameters
JacobianRotation = [[0;0;0],R_1*[0;0;1],R_2*[0;0;1],R_3*[0;0;1]];

omega_0 = [JacobianRotation(:,1),[0;0;0],[0;0;0],[0;0;0]]*...
    q_dot;
omega_1 = [JacobianRotation(:,1:2),[0;0;0],[0;0;0]]*...
    q_dot;
omega_2 = [JacobianRotation(:,1:3),[0;0;0]]*...
    q_dot;
omega_3 = JacobianRotation(:,1:4)*...
    q_dot;

T_10R = 1/2 * transpose(omega_0)*R_1*I_10*transpose(R_1)*omega_0;
T_11R = 1/2 * transpose(omega_1)*R_2*I_11*transpose(R_2)*omega_1;
T_12R = 1/2 * transpose(omega_2)*R_3*I_12*transpose(R_3)*omega_2;
T_13R = 1/2 * transpose(omega_3)*R_E*I_13*transpose(R_E)*omega_3;
```

```
kineticRotationalEnergy = T_l0R + T_l1R + T_l2R + T_l3R;
```

As can be observed we have summed up both the translational and rotational components of our lagrangian, thus our final expression for kinetic energy can be expressed as the expression below.

$$T = \sum_{i=0}^3 \frac{1}{2} \left(m_i \dot{P}_i^T \dot{P}_i + \omega_i^T I_i \omega_i \right)$$

The potential energy is a simpler aspect to estimate, by taking the mass, gravity vector, and position of the COM for each link, our calculation then becomes the expression below;

$$U = \sum_{i=0}^3 (m_i g^T P_i)$$

```
U_l0 = -m_l0*transpose(g)*transl(p_l0);
U_l1 = -m_l1*transpose(g)*transl(p_l1);
U_l2 = -m_l2*transpose(g)*transl(p_l2);
U_l3 = -m_l3*transpose(g)*transl(p_l3);

potentialEnergy = U_l0+U_l1+U_l2+U_l3;

%Final Lagrangian Expression
TotalKineticEnergy = kineticTranslationalEnergy + ...
    kineticRotationalEnergy;

Lagrangian = TotalKineticEnergy - potentialEnergy;
```

The final expression then becomes $L = T - U$.

The next step as outlined below performs the differential steps required to estimate the torque of each actuator. This first involves partially deriving our lagrangian by each actuator's velocity followed by a derivation of time. The potential energy is also derived by the position of the relevant actuator.

$$\tau_i = \frac{d}{dt} \left(\frac{\partial}{\partial \dot{q}_i} L \right) - \frac{\partial}{\partial q_i} U$$

```
%These are the partial differentiation by m positions
m0SubtractionComponent = diff(Lagrangian,q_0);
m1SubtractionComponent = diff(Lagrangian,q_1);
m2SubtractionComponent = diff(Lagrangian,q_2);
m3SubtractionComponent = diff(Lagrangian,q_3);

%These are the partial differentiation by m speed,
% the time derivative
%needs to be taken from these components
syms m0_L_diff(q_0,q_1,q_2,q_3,q_dot_0,q_dot_1,q_dot_2,q_dot_3);
syms m1_L_diff(q_0,q_1,q_2,q_3,q_dot_0,q_dot_1,q_dot_2,q_dot_3);
syms m2_L_diff(q_0,q_1,q_2,q_3,q_dot_0,q_dot_1,q_dot_2,q_dot_3);
```

```

syms m3_L_diff(q_0,q_1,q_2,q_3,q_dot_0,q_dot_1,q_dot_2,q_dot_3);

m0_L_diff(q_0,q_1,q_2,q_3,q_dot_0,q_dot_1,...
    q_dot_2,q_dot_3) = diff(Lagrangian,q_dot_0);
m1_L_diff(q_0,q_1,q_2,q_3,q_dot_0,q_dot_1,...
    q_dot_2,q_dot_3) = diff(Lagrangian,q_dot_1);
m2_L_diff(q_0,q_1,q_2,q_3,q_dot_0,q_dot_1,...
    q_dot_2,q_dot_3) = diff(Lagrangian,q_dot_2);
m3_L_diff(q_0,q_1,q_2,q_3,q_dot_0,q_dot_1,...
    q_dot_2,q_dot_3) = diff(Lagrangian,q_dot_3);

syms t q_dot_0(t) q_0(t) q_dot_1(t) q_1(t) q_dot_2(t) q_2(t);
syms q_dot_3(t) q_3(t);
%take the time derivatives
timeDer0eq = m0_L_diff(q_0(t),q_1(t),q_2(t),q_3(t),...
    q_dot_0(t),q_dot_1(t),q_dot_2(t),q_dot_3(t));
timeDer1eq = m1_L_diff(q_0(t),q_1(t),q_2(t),q_3(t),...
    q_dot_0(t),q_dot_1(t),q_dot_2(t),q_dot_3(t));
timeDer2eq = m2_L_diff(q_0(t),q_1(t),q_2(t),q_3(t),...
    q_dot_0(t),q_dot_1(t),q_dot_2(t),q_dot_3(t));
timeDer3eq = m3_L_diff(q_0(t),q_1(t),q_2(t),q_3(t),...
    q_dot_0(t),q_dot_1(t),q_dot_2(t),q_dot_3(t));
timeDer0 = diff(timeDer0eq,t);
timeDer1 = diff(timeDer1eq,t);
timeDer2 = diff(timeDer2eq,t);
timeDer3 = diff(timeDer3eq,t);
out_tau0 = simplify(timeDer0 - m0SubtractionComponent);
out_tau1 = simplify(timeDer1 - m1SubtractionComponent);
out_tau2 = simplify(timeDer2 - m2SubtractionComponent);
out_tau3 = simplify(timeDer3 - m3SubtractionComponent);
syms q0 q1 q2 q3 qd0 qd1 qd2 qd3 qdd0 qdd1 qdd2 qdd3;
%Declare some variables that represent the time functions for
%replacement in the symbolic equation
qdd0_af = diff(q_dot_0(t), t);
qdd1_af = diff(q_dot_1(t), t);
qdd2_af = diff(q_dot_2(t), t);
qdd3_af = diff(q_dot_3(t), t);

qd0_af = diff(q_0(t), t);
qd1_af = diff(q_1(t), t);
qd2_af = diff(q_2(t), t);
qd3_af = diff(q_3(t), t);

qd0_af2 = q_dot_0(t);
qd1_af2 = q_dot_1(t);
qd2_af2 = q_dot_2(t);
qd3_af2 = q_dot_3(t);

q0_af = q_0(t);
q1_af = q_1(t);

```

```

q2_af = q_2(t);
q3_af = q_3(t);

syms q_0 q_1 q_2 q_3
%remove the time based variables and replace with variables
%for position, velocity and acceleration
srcVariables = {qdd0_af qdd1_af qdd2_af qdd3_af...
    qd0_af qd1_af qd2_af qd3_af...
    qd0_af2 qd1_af2 qd2_af2...
    qd3_af2 q_dot_0 q_dot_1 q_dot_2 q_dot_3...
    q0_af q1_af q2_af q3_af q_0 q_1 q_2 q_3};
replaceList = {qdd0 qdd1 qdd2 qdd3 qd0 qd1 qd2 qd3...
    qd0 qd1 qd2 qd3 qd0 qd1 qd2 qd3 q0 q1 q2 q3 q0 q1 q2 q3};

```

The previous segments of code performed several functions including performing different differential operations among others.

```

tau0 = subs(out_tau0, srcVariables, replaceList);
tau1 = subs(out_tau1, srcVariables, replaceList);
tau2 = subs(out_tau2, srcVariables, replaceList);
tau3 = subs(out_tau3, srcVariables, replaceList);
tau_expressions = [tau0,tau1,tau2,tau3];

```

Now we create the robotic toolbox SerialLink object and validate the lagrangian matches the toolbox. We input the same dynamic parameters that estimate each component.

```

prisMn = 0;
prisMx = 0.12;
links(1).jType = 'prismatic';
links(1).THETA = sym_pi;
links(1).DISTANCE = 0;
links(1).A_DISTANCE = 0;
links(1).ALPHA = sym_pi/2;
links(1).QLIM = [prisMn,prisMx];
links(1).MASS = m_l0;
links(1).COM = p_l0_model;
links(1).INERTIATENSOR = I_l0;
bioGrp(1) = Link(links(1).jType, 'theta', links(1).THETA, 'a',...
    links(1).A_DISTANCE, 'alpha', links(1).ALPHA,...
    'qlim',links(1).QLIM,'m',links(1).MASS,...
    'r',links(1).COM,'I',links(1).INERTIATENSOR,'standard');

links(2).jType = 'revolute';
links(2).THETA = 0;
links(2).DISTANCE = d_1;
links(2).A_DISTANCE = a_1;
links(2).ALPHA = sym_pi/2;
links(2).QLIM = [0.872665,2.26893];
links(2).MASS = m_l1;

```



```

links(2).COM = p_l1_model;
links(2).INERTIATENSOR = I_l1;
bioGrp(2) = Link(links(2).jType, 'd', links(2).DISTANCE, 'a',...
    links(2).A_DISTANCE, 'alpha', links(2).ALPHA,...
    'qlim',links(2).QLIM,'m',links(2).MASS,...
    'r',links(2).COM,'I',links(2).INERTIATENSOR,'standard');

links(3).jType = 'revolute';
links(3).THETA = 0;
links(3).DISTANCE = 0;
links(3).A_DISTANCE = a_2;
links(3).ALPHA = 0;
links(3).QLIM = [0,sym_pi/2];
links(3).MASS = m_l2;
links(3).COM = p_l2_model;
links(3).INERTIATENSOR = I_l2;

bioGrp(3) = Link(links(3).jType, 'd', links(3).DISTANCE, 'a',...
    links(3).A_DISTANCE, 'alpha', links(3).ALPHA,...
    'qlim',links(3).QLIM,'m',links(3).MASS,...
    'r',links(3).COM,'I',links(3).INERTIATENSOR,'standard');

links(4).jType = 'revolute';
links(4).THETA = 0;
links(4).DISTANCE = 0;
links(4).A_DISTANCE = a_3;
links(4).ALPHA = -sym_pi/2;
links(4).QLIM = [-1.74533,0.017453];
links(4).MASS = m_l3;
links(4).COM = p_l3_model;
links(4).INERTIATENSOR = I_l3;

bioGrp(4) = Link(links(4).jType, 'd', links(4).DISTANCE, 'a',...
    links(4).A_DISTANCE, 'alpha', links(4).ALPHA,...
    'qlim',links(4).QLIM,'m',links(4).MASS,'r',...
    links(4).COM,'I',links(4).INERTIATENSOR,'standard');

%This creates our mechanical system in the robotics toolbox,
% bearing in mind the gravity vector is inverted due to
% the toolbox's design and input
%parameters
gripper = SerialLink(bioGrp,'name','BM_endeffector','base',...
    trotz(sym_pi/2)*trotx(sym_pi/2),'gravity',-g)

```

```
gripper =
```

```

BM_endeffector:: 4 axis, PRRR, stdDH, slowRNE, Symbolic
+-----+-----+-----+-----+-----+
| j |   theta |     d |     a |   alpha |  offset |
+-----+-----+-----+-----+-----+
| 1 |     pi |    q1 |     0 |   pi/2 |     0 |

```

2	q2	d_1	a_1	pi/2	0
3	q3	0	a_2	0	0
4	q4	0	a_3	-pi/2	0

base: t = (0, 0, 0), RPY/xyz = (0, 90, 90) deg

```
%Get the m torques from the toolboxes recursive
% Newton Euler algorithm
rne = gripper.rne([q0 q1 q2 q3],[qd0 qd1 qd2 qd3],...
[qdd0 qdd1 qdd2 qdd3]);
```

Validate our equation is equivalent to the rne toolbox calculation as done below;

```
isequal(tau0,rne(1))
```

```
ans = logical
     1
```

```
isequal(tau1,rne(2))
```

```
ans = logical
     1
```

```
isequal(simplify(tau2),simplify(rne(3)))
```

```
ans = logical
     1
```

```
isequal(tau3,rne(4))
```

```
ans = logical
     1
```

Next, we create the coriolis, inertia and gravity terms to compare these results to our lagrangian and validate all torque estimation methods are equivalent.

These matrices can be applied in the following expression to estimate the forces and torques in which Q represents the torque/force vector, remembering that this representation omits the friction term.

$$Q = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q)$$

```
%Generate the inertia matrix with the toolbox
inertiaMatrix = gripper.inertia([q0 q1 q2 q3]);
%Generate the coriolis matrix with the toolbox
coriolisMatrix = gripper.coriolis([q0 q1 q2 q3],[qd0 qd1 qd2 qd3]);
%Generate the gravity term component with the toolbox
gravload = gripper.gravload([q0 q1 q2 q3]);

%Simplify the equations from our lagrangian expressions
Q1 = simplify(tau0);
Q2 = simplify(tau1);
Q3 = simplify(factor(tau2));
Q4 = simplify(factor(tau3));
%Create the lagrangian Q vector
```

```

Q = [Q1;Q2;Q3(1)*Q3(2);Q4(1)*Q4(2)];

%From the inertia and coriolis matrices with the gravity term,
% create a consise descriptor of the lagrangian
Q_evaluation = inertiaMatrix*[qdd0;qdd1;qdd2;qdd3] + ...
    coriolisMatrix*[qd0;qd1;qd2;qd3] + transpose(gravload);

Q_e1 = simplify(Q_evaluation(1));
Q_e2 = simplify(Q_evaluation(2));
Q_e3 = simplify(factor(Q_evaluation(3)));
Q_e4 = simplify(factor(Q_evaluation(4)));
Q_eval = [Q_e1;Q_e2;Q_e3(1)*Q_e3(2);Q_e4(1)*Q_e4(2)];

```

Finally, lets validate that our equations are equivilant with the following evaluations below;

```
isequal(Q(1),Q_eval(1))
```

```
ans = logical
     1
```

```
isequal(Q(2),Q_eval(2))
```

```
ans = logical
     1
```

```
isequal(Q(3),Q_eval(3))
```

```
ans = logical
     1
```

```
isequal(Q(4),Q_eval(4))
```

```
ans = logical
     1
```

Finally, our data can be saved to a matlab mat file holding the symbolic data, done below, we also save our equations to a latex format;

```

latex_tau0 = latex(Q1);
latex_tau1 = latex(Q2);
latex_tau2 = latex(simplify(Q3(1)*Q3(2)));
latex_tau3 = latex(simplify(Q4(1)*Q4(2)));
save('symbolicData');

```

B.0.5 Simplified Tau Expressions

This section details the final equations from the Lagrangian simplified with MATLAB code. First, this thesis provides an outline of the system variables.

- q_i - The position of joint i
- qd_i - The velocity of joint i
- qdd_i - The acceleration of joint i
- a_i - The the a transform of the DH parameters of row i
- d_i - The the d transform of the DH parameters of row i
- τ_i - The torque or force required at from joint i
- m_{li} - The mass of link i
- p_{li} - The CoM translation of link i . This transformation represents the end of the i th row of the DH parameters. The following terms of x , y and z represent the direction for the single variable.
- I_{li} - The inertia tensor about the CoM of link i . The sequential terms of xx , yy , zz , xy , xz and yz denote the respective components of the inertia tensor.

τ_0 Calculation

$$\begin{aligned}
\tau_0 = & m_{10} qdd_0 - g_x m_{11} - g_x m_{12} - g_x m_{13} - g_x m_{10} + m_{11} qdd_0 + m_{12} qdd_0 + m_{13} qdd_0 + \\
& a_1 m_{11} qdd_1 \cos(q_1) + a_1 m_{12} qdd_1 \cos(q_1) + a_1 m_{13} qdd_1 \cos(q_1) + m_{11} p_{11x} qdd_1 \cos(q_1) + \\
& m_{11} p_{11z} qdd_1 \sin(q_1) + m_{12} p_{12z} qdd_1 \sin(q_1) - m_{13} p_{13y} qdd_1 \sin(q_1) + m_{11} p_{11z} qd_1^2 \cos(q_1) + \\
& m_{12} p_{12z} qd_1^2 \cos(q_1) - m_{13} p_{13y} qd_1^2 \cos(q_1) - a_1 m_{11} qd_1^2 \sin(q_1) - a_1 m_{12} qd_1^2 \sin(q_1) - \\
& a_1 m_{13} qd_1^2 \sin(q_1) - m_{11} p_{11x} qd_1^2 \sin(q_1) + a_2 m_{12} qdd_1 \cos(q_1) \cos(q_2) + a_2 m_{13} qdd_1 \cos(q_1) \cos(q_2) + \\
& m_{12} p_{12x} qdd_1 \cos(q_1) \cos(q_2) - m_{12} p_{12y} qdd_1 \cos(q_1) \sin(q_2) - m_{12} p_{12y} qdd_2 \cos(q_2) \sin(q_1) - \\
& a_2 m_{12} qdd_2 \sin(q_1) \sin(q_2) - a_2 m_{13} qdd_2 \sin(q_1) \sin(q_2) - m_{12} p_{12x} qdd_2 \sin(q_1) \sin(q_2) - \\
& a_2 m_{12} qd_1^2 \cos(q_2) \sin(q_1) - a_2 m_{12} qd_2^2 \cos(q_2) \sin(q_1) - a_2 m_{13} qd_1^2 \cos(q_2) \sin(q_1) - \\
& a_2 m_{13} qd_2^2 \cos(q_2) \sin(q_1) - m_{12} p_{12x} qd_1^2 \cos(q_2) \sin(q_1) - m_{12} p_{12x} qd_2^2 \cos(q_2) \sin(q_1) + \\
& m_{12} p_{12y} qd_1^2 \sin(q_1) \sin(q_2) + m_{12} p_{12y} qd_2^2 \sin(q_1) \sin(q_2) + a_3 m_{13} qdd_1 \cos(q_1) \cos(q_2) \cos(q_3) + \\
& m_{13} p_{13x} qdd_1 \cos(q_1) \cos(q_2) \cos(q_3) - m_{13} p_{13z} qdd_1 \cos(q_1) \cos(q_2) \sin(q_3) - \\
& m_{13} p_{13z} qdd_1 \cos(q_1) \cos(q_3) \sin(q_2) - m_{13} p_{13z} qdd_2 \cos(q_2) \cos(q_3) \sin(q_1) - \\
& m_{13} p_{13z} qdd_3 \cos(q_2) \cos(q_3) \sin(q_1) - a_3 m_{13} qdd_1 \cos(q_1) \sin(q_2) \sin(q_3) -
\end{aligned}$$

$$\begin{array}{rcl}
a_3 m_{13} q d d_2 \cos(q_2) \sin(q_1) \sin(q_3) & - & a_3 m_{13} q d d_2 \cos(q_3) \sin(q_1) \sin(q_2) & - \\
a_3 m_{13} q d d_3 \cos(q_2) \sin(q_1) \sin(q_3) & - & a_3 m_{13} q d d_3 \cos(q_3) \sin(q_1) \sin(q_2) & - \\
m_{13} p_{13x} q d d_1 \cos(q_1) \sin(q_2) \sin(q_3) & - & m_{13} p_{13x} q d d_2 \cos(q_2) \sin(q_1) \sin(q_3) & - \\
m_{13} p_{13x} q d d_2 \cos(q_3) \sin(q_1) \sin(q_2) & - & m_{13} p_{13x} q d d_3 \cos(q_2) \sin(q_1) \sin(q_3) & - \\
m_{13} p_{13x} q d d_3 \cos(q_3) \sin(q_1) \sin(q_2) & + & m_{13} p_{13z} q d d_2 \sin(q_1) \sin(q_2) \sin(q_3) & + \\
m_{13} p_{13z} q d d_3 \sin(q_1) \sin(q_2) \sin(q_3) & - & a_3 m_{13} q d_1^2 \cos(q_2) \cos(q_3) \sin(q_1) & - \\
a_3 m_{13} q d_2^2 \cos(q_2) \cos(q_3) \sin(q_1) & - & a_3 m_{13} q d_3^2 \cos(q_2) \cos(q_3) \sin(q_1) & - \\
m_{13} p_{13x} q d_1^2 \cos(q_2) \cos(q_3) \sin(q_1) & - & m_{13} p_{13x} q d_2^2 \cos(q_2) \cos(q_3) \sin(q_1) & - \\
m_{13} p_{13x} q d_3^2 \cos(q_2) \cos(q_3) \sin(q_1) & + & m_{13} p_{13z} q d_1^2 \cos(q_2) \sin(q_1) \sin(q_3) & + \\
m_{13} p_{13z} q d_1^2 \cos(q_3) \sin(q_1) \sin(q_2) & + & m_{13} p_{13z} q d_2^2 \cos(q_2) \sin(q_1) \sin(q_3) & + \\
m_{13} p_{13z} q d_2^2 \cos(q_3) \sin(q_1) \sin(q_2) & + & m_{13} p_{13z} q d_3^2 \cos(q_2) \sin(q_1) \sin(q_3) & + \\
m_{13} p_{13z} q d_3^2 \cos(q_3) \sin(q_1) \sin(q_2) & + & a_3 m_{13} q d_1^2 \sin(q_1) \sin(q_2) \sin(q_3) & + \\
a_3 m_{13} q d_2^2 \sin(q_1) \sin(q_2) \sin(q_3) & + & a_3 m_{13} q d_3^2 \sin(q_1) \sin(q_2) \sin(q_3) & + \\
m_{13} p_{13x} q d_1^2 \sin(q_1) \sin(q_2) \sin(q_3) & + & m_{13} p_{13x} q d_2^2 \sin(q_1) \sin(q_2) \sin(q_3) & + \\
m_{13} p_{13x} q d_3^2 \sin(q_1) \sin(q_2) \sin(q_3) & - & 2 m_{12} p_{12y} q d_1 q d_2 \cos(q_1) \cos(q_2) & - \\
2 a_2 m_{12} q d_1 q d_2 \cos(q_1) \sin(q_2) - 2 a_2 m_{13} q d_1 q d_2 \cos(q_1) \sin(q_2) - 2 m_{12} p_{12x} q d_1 q d_2 \cos(q_1) \sin(q_2) & - & & - \\
2 m_{13} p_{13z} q d_1 q d_2 \cos(q_1) \cos(q_2) \cos(q_3) & - & 2 m_{13} p_{13z} q d_1 q d_3 \cos(q_1) \cos(q_2) \cos(q_3) & - \\
2 a_3 m_{13} q d_1 q d_2 \cos(q_1) \cos(q_2) \sin(q_3) & - & 2 a_3 m_{13} q d_1 q d_2 \cos(q_1) \cos(q_3) \sin(q_2) & - \\
2 a_3 m_{13} q d_1 q d_3 \cos(q_1) \cos(q_2) \sin(q_3) & - & 2 a_3 m_{13} q d_1 q d_3 \cos(q_1) \cos(q_3) \sin(q_2) & - \\
2 a_3 m_{13} q d_2 q d_3 \cos(q_2) \cos(q_3) \sin(q_1) & - & 2 m_{13} p_{13x} q d_1 q d_2 \cos(q_1) \cos(q_2) \sin(q_3) & - \\
2 m_{13} p_{13x} q d_1 q d_2 \cos(q_1) \cos(q_3) \sin(q_2) & - & 2 m_{13} p_{13x} q d_1 q d_3 \cos(q_1) \cos(q_2) \sin(q_3) & - \\
2 m_{13} p_{13x} q d_1 q d_3 \cos(q_1) \cos(q_3) \sin(q_2) & - & 2 m_{13} p_{13x} q d_2 q d_3 \cos(q_2) \cos(q_3) \sin(q_1) & + \\
2 m_{13} p_{13z} q d_1 q d_2 \cos(q_1) \sin(q_2) \sin(q_3) & + & 2 m_{13} p_{13z} q d_1 q d_3 \cos(q_1) \sin(q_2) \sin(q_3) & + \\
2 m_{13} p_{13z} q d_2 q d_3 \cos(q_2) \sin(q_1) \sin(q_3) & + & 2 m_{13} p_{13z} q d_2 q d_3 \cos(q_3) \sin(q_1) \sin(q_2) & + \\
2 a_3 m_{13} q d_2 q d_3 \sin(q_1) \sin(q_2) \sin(q_3) + 2 m_{13} p_{13x} q d_2 q d_3 \sin(q_1) \sin(q_2) \sin(q_3) & & & +
\end{array}$$

τ_1 Calculation

$$\begin{aligned}
\tau_1 = & \frac{I_{12xx} q d d_1}{2} + I_{11yy} q d d_1 + \frac{I_{13xx} q d d_1}{2} + \frac{I_{12yy} q d d_1}{2} + \frac{I_{13zz} q d d_1}{2} + m_{11} p_{11x}^2 q d d_1 + m_{11} p_{11z}^2 q d d_1 + \\
& \frac{m_{12} p_{12x}^2 q d d_1}{2} + \frac{m_{12} p_{12y}^2 q d d_1}{2} + m_{12} p_{12z}^2 q d d_1 + \frac{m_{13} p_{13x}^2 q d d_1}{2} + m_{13} p_{13y}^2 q d d_1 + \frac{m_{13} p_{13z}^2 q d d_1}{2} - \\
& I_{13xy} q d_2^2 \cos(q_2 + q_3) - I_{13xy} q d_3^2 \cos(q_2 + q_3) + I_{13yz} q d_2^2 \sin(q_2 + q_3) + I_{13yz} q d_3^2 \sin(q_2 + q_3) - \\
& \frac{I_{12xx} q d d_1 \cos(2q_2)}{2} + \frac{I_{12yy} q d d_1 \cos(2q_2)}{2} + I_{12xz} q d_2^2 \cos(q_2) + I_{12xy} q d d_1 \sin(2q_2) - I_{12yz} q d_2^2 \sin(q_2) - \\
& \frac{I_{13xx} q d d_1 \cos(2q_2 + 2q_3)}{2} + \frac{I_{13zz} q d d_1 \cos(2q_2 + 2q_3)}{2} + I_{13xz} q d d_1 \sin(2q_2 + 2q_3) - I_{13yz} q d d_2 \cos(q_2 + q_3) - \\
& I_{13yz} q d d_3 \cos(q_2 + q_3) - I_{13xy} q d d_2 \sin(q_2 + q_3) - I_{13xy} q d d_3 \sin(q_2 + q_3) + I_{12yz} q d d_2 \cos(q_2) + \\
& I_{12xz} q d d_2 \sin(q_2) + a_1^2 m_{11} q d d_1 + a_1^2 m_{12} q d d_1 + a_1^2 m_{13} q d d_1 + \frac{a_2^2 m_{12} q d d_1}{2} + \frac{a_2^2 m_{13} q d d_1}{2} + \\
& \frac{a_3^2 m_{13} q d d_1}{2} - \frac{a_3 g_x m_{13} \cos(q_2 - q_1 + q_3)}{2} + 2 a_1 m_{11} p_{11x} q d d_1 + a_2 m_{12} p_{12x} q d d_1 + a_3 m_{13} p_{13x} q d d_1 +
\end{aligned}$$

$$\begin{aligned}
& \frac{a_3 m_{13} qdd_0 \cos(q_2 - q_1 + q_3)}{2} - \frac{g_x m_{13} p_{13x} \cos(q_2 - q_1 + q_3)}{2} + \frac{g_y m_{13} p_{13z} \cos(q_2 - q_1 + q_3)}{2} + \frac{a_3 g_y m_{13} \sin(q_2 - q_1 + q_3)}{2} + \\
& \frac{m_{13} p_{13x} qdd_0 \cos(q_2 - q_1 + q_3)}{2} + \frac{g_x m_{13} p_{13z} \sin(q_2 - q_1 + q_3)}{2} + \frac{g_y m_{13} p_{13x} \sin(q_2 - q_1 + q_3)}{2} - 2 I_{13xy} qd_2 qd_3 \cos(q_2 + q_3) + \\
& \frac{a_2^2 m_{12} qdd_1 \cos(2q_2)}{2} + \frac{a_2^2 m_{13} qdd_1 \cos(2q_2)}{2} - \frac{m_{13} p_{13z} qdd_0 \sin(q_2 - q_1 + q_3)}{2} - \frac{a_2 g_x m_{12} \cos(q_1 + q_2)}{2} - \frac{a_2 g_x m_{13} \cos(q_1 + q_2)}{2} + \\
& 2 I_{13yz} qd_2 qd_3 \sin(q_2 + q_3) + \frac{m_{12} p_{12x}^2 qdd_1 \cos(2q_2)}{2} - \frac{m_{12} p_{12y}^2 qdd_1 \cos(2q_2)}{2} + \frac{a_2 m_{12} qdd_0 \cos(q_1 + q_2)}{2} + \\
& \frac{a_2 m_{13} qdd_0 \cos(q_1 + q_2)}{2} - \frac{g_x m_{12} p_{12x} \cos(q_1 + q_2)}{2} - \frac{g_y m_{12} p_{12y} \cos(q_1 + q_2)}{2} - \frac{a_2 g_y m_{12} \sin(q_1 + q_2)}{2} - \frac{a_2 g_y m_{13} \sin(q_1 + q_2)}{2} + \\
& \frac{m_{12} p_{12x} qdd_0 \cos(q_1 + q_2)}{2} + \frac{g_x m_{12} p_{12y} \sin(q_1 + q_2)}{2} - \frac{g_y m_{12} p_{12x} \sin(q_1 + q_2)}{2} + \frac{a_3^2 m_{13} qdd_1 \cos(2q_2 + 2q_3)}{2} - \\
& \frac{m_{12} p_{12y} qdd_0 \sin(q_1 + q_2)}{2} - a_1 g_x m_{11} \cos(q_1) - a_1 g_x m_{12} \cos(q_1) - a_1 g_x m_{13} \cos(q_1) + \frac{m_{13} p_{13x}^2 qdd_1 \cos(2q_2 + 2q_3)}{2} - \\
& \frac{m_{13} p_{13z}^2 qdd_1 \cos(2q_2 + 2q_3)}{2} + a_1 m_{11} qdd_0 \cos(q_1) + a_1 m_{12} qdd_0 \cos(q_1) + a_1 m_{13} qdd_0 \cos(q_1) - \\
& g_x m_{11} p_{11x} \cos(q_1) + g_y m_{11} p_{11z} \cos(q_1) + g_y m_{12} p_{12z} \cos(q_1) - g_y m_{13} p_{13y} \cos(q_1) - a_1 g_y m_{11} \sin(q_1) - \\
& a_1 g_y m_{12} \sin(q_1) - a_1 g_y m_{13} \sin(q_1) + m_{11} p_{11x} qdd_0 \cos(q_1) - g_x m_{11} p_{11z} \sin(q_1) - g_x m_{12} p_{12z} \sin(q_1) + \\
& g_x m_{13} p_{13y} \sin(q_1) - g_y m_{11} p_{11x} \sin(q_1) + m_{11} p_{11z} qdd_0 \sin(q_1) + m_{12} p_{12z} qdd_0 \sin(q_1) - \\
& m_{13} p_{13y} qdd_0 \sin(q_1) - \frac{a_2 g_x m_{12} \cos(q_1 - q_2)}{2} - \frac{a_2 g_x m_{13} \cos(q_1 - q_2)}{2} + \frac{a_2 m_{12} qdd_0 \cos(q_1 - q_2)}{2} + \frac{a_2 m_{13} qdd_0 \cos(q_1 - q_2)}{2} - \\
& \frac{g_x m_{12} p_{12x} \cos(q_1 - q_2)}{2} + \frac{g_y m_{12} p_{12y} \cos(q_1 - q_2)}{2} - \frac{a_2 g_y m_{12} \sin(q_1 - q_2)}{2} - \frac{a_2 g_y m_{13} \sin(q_1 - q_2)}{2} - \frac{a_3 g_x m_{13} \cos(q_1 + q_2 + q_3)}{2} + \\
& \frac{m_{12} p_{12x} qdd_0 \cos(q_1 - q_2)}{2} + \frac{a_3 m_{13} qdd_0 \cos(q_1 + q_2 + q_3)}{2} - \frac{g_x m_{12} p_{12y} \sin(q_1 - q_2)}{2} - \frac{g_y m_{12} p_{12x} \sin(q_1 - q_2)}{2} - \\
& \frac{g_x m_{13} p_{13x} \cos(q_1 + q_2 + q_3)}{2} - \frac{g_y m_{13} p_{13z} \cos(q_1 + q_2 + q_3)}{2} - \frac{a_3 g_y m_{13} \sin(q_1 + q_2 + q_3)}{2} + 2 I_{12xy} qd_1 qd_2 \cos(2q_2) + \\
& \frac{m_{12} p_{12y} qdd_0 \sin(q_1 - q_2)}{2} + \frac{m_{13} p_{13x} qdd_0 \cos(q_1 + q_2 + q_3)}{2} + \frac{g_x m_{13} p_{13z} \sin(q_1 + q_2 + q_3)}{2} - \frac{g_y m_{13} p_{13x} \sin(q_1 + q_2 + q_3)}{2} + \\
& I_{12xx} qd_1 qd_2 \sin(2q_2) - I_{2yy} qd_1 qd_2 \sin(2q_2) - \frac{m_{13} p_{13z} qdd_0 \sin(q_1 + q_2 + q_3)}{2} + 2 I_{13xz} qd_1 qd_2 \cos(2q_2 + 2q_3) + \\
& 2 I_{13zx} qd_1 qd_3 \cos(2q_2 + 2q_3) + I_{13xx} qd_1 qd_2 \sin(2q_2 + 2q_3) + I_{13xx} qd_1 qd_3 \sin(2q_2 + 2q_3) - \\
& I_{13zz} qd_1 qd_2 \sin(2q_2 + 2q_3) - I_{13zz} qd_1 qd_3 \sin(2q_2 + 2q_3) - m_{12} p_{12x} p_{12z} qdd_2 \sin(q_2) + \\
& a_2 a_3 m_{13} qdd_1 \cos(2q_2 + q_3) + a_2 m_{13} p_{13x} qdd_1 \cos(2q_2 + q_3) + a_3 m_{13} p_{13y} qd_2^2 \cos(q_2 + q_3) + \\
& a_3 m_{13} p_{13y} qd_3^2 \cos(q_2 + q_3) + m_{13} p_{13x} p_{13y} qd_2^2 \cos(q_2 + q_3) + m_{13} p_{13x} p_{13y} qd_3^2 \cos(q_2 + q_3) - \\
& a_2 m_{13} p_{13z} qdd_1 \sin(2q_2 + q_3) - m_{13} p_{13y} p_{13z} qd_2^2 \sin(q_2 + q_3) - m_{13} p_{13y} p_{13z} qd_3^2 \sin(q_2 + q_3) + \\
& a_2 m_{12} p_{12x} qdd_1 \cos(2q_2) - a_2 m_{12} p_{12z} qd_2^2 \cos(q_2) + a_2 m_{13} p_{13y} qd_2^2 \cos(q_2) - m_{12} p_{12x} p_{12z} qd_2^2 \cos(q_2) - \\
& a_2 m_{12} p_{12y} qdd_1 \sin(2q_2) - m_{12} p_{12x} p_{12y} qdd_1 \sin(2q_2) + m_{12} p_{12y} p_{12z} qd_2^2 \sin(q_2) + \\
& a_3 m_{13} p_{13x} qdd_1 \cos(2q_2 + 2q_3) - a_3 m_{13} p_{13z} qdd_1 \sin(2q_2 + 2q_3) - m_{13} p_{13x} p_{13z} qdd_1 \sin(2q_2 + 2q_3) + \\
& 2 a_1 a_3 m_{13} qdd_1 \cos(q_2 + q_3) - a_2^2 m_{12} qd_1 qd_2 \sin(2q_2) - a_2^2 m_{13} qd_1 qd_2 \sin(2q_2) + \\
& 2 a_1 m_{13} p_{13x} qdd_1 \cos(q_2 + q_3) - m_{12} p_{12x}^2 qd_1 qd_2 \sin(2q_2) + m_{12} p_{12y}^2 qd_1 qd_2 \sin(2q_2) + \\
& m_{13} p_{13y} p_{13z} qdd_2 \cos(q_2 + q_3) + m_{13} p_{13y} p_{13z} qdd_3 \cos(q_2 + q_3) - 2 a_1 m_{13} p_{13z} qdd_1 \sin(q_2 + q_3) + \\
& a_3 m_{13} p_{13y} qdd_2 \sin(q_2 + q_3) + a_3 m_{13} p_{13y} qdd_3 \sin(q_2 + q_3) + m_{13} p_{13x} p_{13y} qdd_2 \sin(q_2 + q_3) + \\
& m_{13} p_{13x} p_{13y} qdd_3 \sin(q_2 + q_3) + 2 a_1 a_2 m_{12} qdd_1 \cos(q_2) + 2 a_1 a_2 m_{13} qdd_1 \cos(q_2) + \\
& a_2 a_3 m_{13} qdd_1 \cos(q_3) - a_3^2 m_{13} qd_1 qd_2 \sin(2q_2 + 2q_3) - a_3^2 m_{13} qd_1 qd_3 \sin(2q_2 + 2q_3) + \\
& 2 a_1 m_{12} p_{12x} qdd_1 \cos(q_2) + a_2 m_{13} p_{13x} qdd_1 \cos(q_3) - m_{13} p_{13x}^2 qd_1 qd_2 \sin(2q_2 + 2q_3) - \\
& m_{13} p_{13x}^2 qd_1 qd_3 \sin(2q_2 + 2q_3) + m_{13} p_{13z}^2 qd_1 qd_2 \sin(2q_2 + 2q_3) + m_{13} p_{13z}^2 qd_1 qd_3 \sin(2q_2 + 2q_3) - \\
& m_{12} p_{12y} p_{12z} qdd_2 \cos(q_2) - 2 a_1 m_{12} p_{12y} qdd_1 \sin(q_2) - a_2 m_{12} p_{12z} qdd_2 \sin(q_2) + a_2 m_{13} p_{13y} qdd_2 \sin(q_2) - \\
& a_2 m_{13} p_{13z} qdd_1 \sin(q_3) - 2 a_1 m_{13} p_{13z} qd_1 qd_2 \cos(q_2 + q_3) - 2 a_1 m_{13} p_{13z} qd_1 qd_3 \cos(q_2 + q_3) +
\end{aligned}$$

$$\begin{aligned}
& 2 a_3 m_{13} p_{13y} qd_2 qd_3 \cos(q_2 + q_3) - 2 a_1 a_3 m_{13} qd_1 qd_2 \sin(q_2 + q_3) - 2 a_1 a_3 m_{13} qd_1 qd_3 \sin(q_2 + q_3) + \\
& 2 m_{13} p_{13x} p_{13y} qd_2 qd_3 \cos(q_2 + q_3) - 2 a_1 m_{13} p_{13x} qd_1 qd_2 \sin(q_2 + q_3) - 2 a_1 m_{13} p_{13x} qd_1 qd_3 \sin(q_2 + q_3) - \\
& 2 m_{13} p_{13y} p_{13z} qd_2 qd_3 \sin(q_2 + q_3) - 2 a_1 m_{12} p_{12y} qd_1 qd_2 \cos(q_2) - a_2 m_{13} p_{13z} qd_1 qd_3 \cos(q_3) - \\
& 2 a_1 a_2 m_{12} qd_1 qd_2 \sin(q_2) - 2 a_1 a_2 m_{13} qd_1 qd_2 \sin(q_2) - a_2 a_3 m_{13} qd_1 qd_3 \sin(q_3) - \\
& 2 a_1 m_{12} p_{12x} qd_1 qd_2 \sin(q_2) - a_2 m_{13} p_{13x} qd_1 qd_3 \sin(q_3) - 2 a_2 m_{13} p_{13z} qd_1 qd_2 \cos(2q_2 + q_3) - \\
& a_2 m_{13} p_{13z} qd_1 qd_3 \cos(2q_2 + q_3) - 2 a_2 a_3 m_{13} qd_1 qd_2 \sin(2q_2 + q_3) - a_2 a_3 m_{13} qd_1 qd_3 \sin(2q_2 + q_3) - \\
& 2 a_2 m_{13} p_{13x} qd_1 qd_2 \sin(2q_2 + q_3) - a_2 m_{13} p_{13x} qd_1 qd_3 \sin(2q_2 + q_3) - 2 a_2 m_{12} p_{12y} qd_1 qd_2 \cos(2q_2) - \\
& 2 m_{12} p_{12x} p_{12y} qd_1 qd_2 \cos(2q_2) - 2 a_2 m_{12} p_{12x} qd_1 qd_2 \sin(2q_2) - 2 a_3 m_{13} p_{13z} qd_1 qd_2 \cos(2q_2 + 2q_3) - \\
& 2 a_3 m_{13} p_{13z} qd_1 qd_3 \cos(2q_2 + 2q_3) - 2 m_{13} p_{13x} p_{13z} qd_1 qd_2 \cos(2q_2 + 2q_3) - \\
& 2 m_{13} p_{13x} p_{13z} qd_1 qd_3 \cos(2q_2 + 2q_3) - 2 a_3 m_{13} p_{13x} qd_1 qd_2 \sin(2q_2 + 2q_3) - \\
& 2 a_3 m_{13} p_{13x} qd_1 qd_3 \sin(2q_2 + 2q_3)
\end{aligned}$$

τ_2 Calculation

$$\begin{aligned}
\tau_2 = & I_{13yy} qdd_2 + I_{13yy} qdd_3 + I_{12zz} qdd_2 + m_{12} p_{12x}^2 qdd_2 + m_{12} p_{12y}^2 qdd_2 + m_{13} p_{13x}^2 qdd_2 + \\
& m_{13} p_{13x}^2 qdd_3 + m_{13} p_{13z}^2 qdd_2 + m_{13} p_{13z}^2 qdd_3 - I_{12xy} qd_1^2 \cos(2q_2) - \frac{I_{12xx} qd_1^2 \sin(2q_2)}{2} + \frac{I_{12yy} qd_1^2 \sin(2q_2)}{2} - \\
& I_{13yz} qdd_1 \cos(q_2 + q_3) - I_{13xy} qdd_1 \sin(q_2 + q_3) - I_{13xz} qd_1^2 \cos(2q_2 + 2q_3) - \frac{I_{13xx} qd_1^2 \sin(2q_2 + 2q_3)}{2} + \\
& \frac{I_{13zz} qd_1^2 \sin(2q_2 + 2q_3)}{2} + I_{12yz} qdd_1 \cos(q_2) + I_{12xz} qdd_1 \sin(q_2) + a_2^2 m_{12} qdd_2 + a_2^2 m_{13} qdd_2 + a_3^2 m_{13} qdd_2 + \\
& a_3^2 m_{13} qdd_3 + \frac{a_3 g_x m_{13} \cos(q_2 - q_1 + q_3)}{2} + 2 a_2 m_{12} p_{12x} qdd_2 + 2 a_3 m_{13} p_{13x} qdd_2 + 2 a_3 m_{13} p_{13x} qdd_3 - \\
& \frac{a_3 m_{13} qdd_0 \cos(q_2 - q_1 + q_3)}{2} + \frac{g_x m_{13} p_{13x} \cos(q_2 - q_1 + q_3)}{2} - \frac{g_y m_{13} p_{13z} \cos(q_2 - q_1 + q_3)}{2} - \frac{a_3 g_y m_{13} \sin(q_2 - q_1 + q_3)}{2} - \\
& \frac{m_{13} p_{13x} qdd_0 \cos(q_2 - q_1 + q_3)}{2} - \frac{g_x m_{13} p_{13z} \sin(q_2 - q_1 + q_3)}{2} - \frac{g_y m_{13} p_{13x} \sin(q_2 - q_1 + q_3)}{2} + \frac{m_{13} p_{13z} qdd_0 \sin(q_2 - q_1 + q_3)}{2} - \\
& \frac{a_2 g_x m_{12} \cos(q_1 + q_2)}{2} - \frac{a_2 g_x m_{13} \cos(q_1 + q_2)}{2} - a_3 g_z m_{13} \cos(q_2 + q_3) + \frac{a_2 m_{12} qdd_0 \cos(q_1 + q_2)}{2} + \\
& \frac{a_2 m_{13} qdd_0 \cos(q_1 + q_2)}{2} - \frac{g_x m_{12} p_{12x} \cos(q_1 + q_2)}{2} - \frac{g_y m_{12} p_{12y} \cos(q_1 + q_2)}{2} - g_z m_{13} p_{13x} \cos(q_2 + q_3) - \\
& \frac{a_2 g_y m_{12} \sin(q_1 + q_2)}{2} - \frac{a_2 g_y m_{13} \sin(q_1 + q_2)}{2} + \frac{m_{12} p_{12x} qdd_0 \cos(q_1 + q_2)}{2} + \frac{g_x m_{12} p_{12y} \sin(q_1 + q_2)}{2} - \frac{g_y m_{12} p_{12x} \sin(q_1 + q_2)}{2} + \\
& g_z m_{13} p_{13z} \sin(q_2 + q_3) - \frac{m_{12} p_{12y} qdd_0 \sin(q_1 + q_2)}{2} - a_2 g_z m_{12} \cos(q_2) - a_2 g_z m_{13} \cos(q_2) - \\
& g_z m_{12} p_{12x} \cos(q_2) + g_z m_{12} p_{12y} \sin(q_2) + \frac{a_2 g_x m_{12} \cos(q_1 - q_2)}{2} + \frac{a_2 g_x m_{13} \cos(q_1 - q_2)}{2} + \frac{a_2^2 m_{12} qd_1^2 \sin(2q_2)}{2} + \\
& \frac{a_2^2 m_{13} qd_1^2 \sin(2q_2)}{2} - \frac{a_2 m_{12} qdd_0 \cos(q_1 - q_2)}{2} - \frac{a_2 m_{13} qdd_0 \cos(q_1 - q_2)}{2} + \frac{g_x m_{12} p_{12x} \cos(q_1 - q_2)}{2} - \frac{g_y m_{12} p_{12y} \cos(q_1 - q_2)}{2} + \\
& \frac{a_2 g_y m_{12} \sin(q_1 - q_2)}{2} + \frac{a_2 g_y m_{13} \sin(q_1 - q_2)}{2} - \frac{a_3 g_x m_{13} \cos(q_1 + q_2 + q_3)}{2} + \frac{m_{12} p_{12x}^2 qd_1^2 \sin(2q_2)}{2} - \frac{m_{12} p_{12y}^2 qd_1^2 \sin(2q_2)}{2} - \\
& \frac{m_{12} p_{12x} qdd_0 \cos(q_1 - q_2)}{2} + \frac{a_3 m_{13} qdd_0 \cos(q_1 + q_2 + q_3)}{2} + \frac{g_x m_{12} p_{12y} \sin(q_1 - q_2)}{2} + \frac{g_y m_{12} p_{12x} \sin(q_1 - q_2)}{2} - \\
& \frac{g_x m_{13} p_{13x} \cos(q_1 + q_2 + q_3)}{2} - \frac{g_y m_{13} p_{13z} \cos(q_1 + q_2 + q_3)}{2} - \frac{a_3 g_y m_{13} \sin(q_1 + q_2 + q_3)}{2} - \frac{m_{12} p_{12y} qdd_0 \sin(q_1 - q_2)}{2} + \\
& \frac{m_{13} p_{13x} qdd_0 \cos(q_1 + q_2 + q_3)}{2} + \frac{g_x m_{13} p_{13z} \sin(q_1 + q_2 + q_3)}{2} - \frac{g_y m_{13} p_{13x} \sin(q_1 + q_2 + q_3)}{2} + \frac{a_3^2 m_{13} qd_1^2 \sin(2q_2 + 2q_3)}{2} - \\
& \frac{m_{13} p_{13z} qdd_0 \sin(q_1 + q_2 + q_3)}{2} + \frac{m_{13} p_{13x}^2 qd_1^2 \sin(2q_2 + 2q_3)}{2} - \frac{m_{13} p_{13z}^2 qd_1^2 \sin(2q_2 + 2q_3)}{2} - m_{12} p_{12x} p_{12z} qdd_1 \sin(q_2) + \\
& a_1 m_{13} p_{13z} qd_1^2 \cos(q_2 + q_3) + a_1 a_3 m_{13} qd_1^2 \sin(q_2 + q_3) + a_1 m_{13} p_{13x} qd_1^2 \sin(q_2 + q_3) + \\
& a_1 m_{12} p_{12y} qd_1^2 \cos(q_2) - a_2 m_{13} p_{13z} qd_3^2 \cos(q_3) + a_1 a_2 m_{12} qd_1^2 \sin(q_2) + a_1 a_2 m_{13} qd_1^2 \sin(q_2) - \\
& a_2 a_3 m_{13} qd_3^2 \sin(q_3) + a_1 m_{12} p_{12x} qd_1^2 \sin(q_2) - a_2 m_{13} p_{13x} qd_3^2 \sin(q_3) + a_2 m_{13} p_{13z} qd_1^2 \cos(2q_2 + q_3) + \\
& a_2 a_3 m_{13} qd_1^2 \sin(2q_2 + q_3) + a_2 m_{13} p_{13x} qd_1^2 \sin(2q_2 + q_3) + a_2 m_{12} p_{12y} qd_1^2 \cos(2q_2) +
\end{aligned}$$

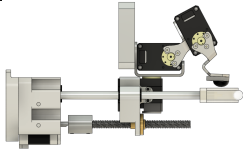
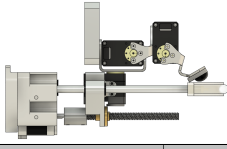
$$\begin{aligned}
& m_{12} p_{12x} p_{12y} qd_1^2 \cos(2q_2) + a_2 m_{12} p_{12x} qd_1^2 \sin(2q_2) + m_{13} p_{13y} p_{13z} qdd_1 \cos(q_2 + q_3) + \\
& a_3 m_{13} p_{13y} qdd_1 \sin(q_2 + q_3) + a_3 m_{13} p_{13z} qd_1^2 \cos(2q_2 + 2q_3) + m_{13} p_{13x} p_{13y} qdd_1 \sin(q_2 + q_3) + \\
& 2a_2 a_3 m_{13} qdd_2 \cos(q_3) + a_2 a_3 m_{13} qdd_3 \cos(q_3) + m_{13} p_{13x} p_{13z} qd_1^2 \cos(2q_2 + 2q_3) + \\
& a_3 m_{13} p_{13x} qd_1^2 \sin(2q_2 + 2q_3) + 2a_2 m_{13} p_{13x} qdd_2 \cos(q_3) + a_2 m_{13} p_{13x} qdd_3 \cos(q_3) - \\
& m_{12} p_{12y} p_{12z} qdd_1 \cos(q_2) - a_2 m_{12} p_{12z} qdd_1 \sin(q_2) + a_2 m_{13} p_{13y} qdd_1 \sin(q_2) - 2a_2 m_{13} p_{13z} qdd_2 \sin(q_3) - \\
& a_2 m_{13} p_{13z} qdd_3 \sin(q_3) - 2a_2 m_{13} p_{13z} qd_2 qd_3 \cos(q_3) - 2a_2 a_3 m_{13} qd_2 qd_3 \sin(q_3) - \\
& 2a_2 m_{13} p_{13x} qd_2 qd_3 \sin(q_3)
\end{aligned}$$

τ_3 Calculation

$$\begin{aligned}
\tau_3 = & I_{3yy} qdd_2 + I_{3yy} qdd_3 + m_{13} p_{13x}^2 qdd_2 + m_{13} p_{13x}^2 qdd_3 + m_{13} p_{13z}^2 qdd_2 + m_{13} p_{13z}^2 qdd_3 - \\
& I_{3yz} qdd_1 \cos(q_2 + q_3) - I_{3xy} qdd_1 \sin(q_2 + q_3) - I_{3xz} qd_1^2 \cos(2q_2 + 2q_3) - \frac{I_{3xx} qd_1^2 \sin(2q_2 + 2q_3)}{2} + \\
& \frac{I_{3zz} qd_1^2 \sin(2q_2 + 2q_3)}{2} + a_3^2 m_{13} qdd_2 + a_3^2 m_{13} qdd_3 + \frac{a_3 g_x m_{13} \cos(q_2 - q_1 + q_3)}{2} + 2a_3 m_{13} p_{13x} qdd_2 + \\
& 2a_3 m_{13} p_{13x} qdd_3 - \frac{a_3 m_{13} qdd_0 \cos(q_2 - q_1 + q_3)}{2} + \frac{g_x m_{13} p_{13x} \cos(q_2 - q_1 + q_3)}{2} - \frac{g_y m_{13} p_{13z} \cos(q_2 - q_1 + q_3)}{2} - \\
& \frac{a_3 g_y m_{13} \sin(q_2 - q_1 + q_3)}{2} - \frac{m_{13} p_{13x} qdd_0 \cos(q_2 - q_1 + q_3)}{2} - \frac{g_x m_{13} p_{13z} \sin(q_2 - q_1 + q_3)}{2} - \frac{g_y m_{13} p_{13x} \sin(q_2 - q_1 + q_3)}{2} + \\
& \frac{m_{13} p_{13z} qdd_0 \sin(q_2 - q_1 + q_3)}{2} - a_3 g_z m_{13} \cos(q_2 + q_3) - g_z m_{13} p_{13x} \cos(q_2 + q_3) + g_z m_{13} p_{13z} \sin(q_2 + q_3) - \\
& \frac{a_3 g_x m_{13} \cos(q_1 + q_2 + q_3)}{2} + \frac{a_3 m_{13} qdd_0 \cos(q_1 + q_2 + q_3)}{2} - \frac{g_x m_{13} p_{13x} \cos(q_1 + q_2 + q_3)}{2} - \frac{g_y m_{13} p_{13z} \cos(q_1 + q_2 + q_3)}{2} - \\
& \frac{a_3 g_y m_{13} \sin(q_1 + q_2 + q_3)}{2} + \frac{m_{13} p_{13x} qdd_0 \cos(q_1 + q_2 + q_3)}{2} + \frac{g_x m_{13} p_{13z} \sin(q_1 + q_2 + q_3)}{2} - \frac{g_y m_{13} p_{13x} \sin(q_1 + q_2 + q_3)}{2} + \\
& \frac{a_3^2 m_{13} qd_1^2 \sin(2q_2 + 2q_3)}{2} - \frac{m_{13} p_{13z} qdd_0 \sin(q_1 + q_2 + q_3)}{2} + \frac{m_{13} p_{13x}^2 qd_1^2 \sin(2q_2 + 2q_3)}{2} - \frac{m_{13} p_{13z}^2 qd_1^2 \sin(2q_2 + 2q_3)}{2} + \\
& a_1 m_{13} p_{13z} qd_1^2 \cos(q_2 + q_3) + a_1 a_3 m_{13} qd_1^2 \sin(q_2 + q_3) + a_1 m_{13} p_{13x} qd_1^2 \sin(q_2 + q_3) + \\
& \frac{a_2 m_{13} p_{13z} qd_1^2 \cos(q_3)}{2} + a_2 m_{13} p_{13z} qd_2^2 \cos(q_3) + \frac{a_2 a_3 m_{13} qd_1^2 \sin(q_3)}{2} + a_2 a_3 m_{13} qd_2^2 \sin(q_3) + \\
& \frac{a_2 m_{13} p_{13x} qd_1^2 \sin(q_3)}{2} + a_2 m_{13} p_{13x} qd_2^2 \sin(q_3) + \frac{a_2 m_{13} p_{13z} qd_1^2 \cos(2q_2 + q_3)}{2} + \frac{a_2 a_3 m_{13} qd_1^2 \sin(2q_2 + q_3)}{2} + \\
& \frac{a_2 m_{13} p_{13x} qd_1^2 \sin(2q_2 + q_3)}{2} + m_{13} p_{13y} p_{13z} qdd_1 \cos(q_2 + q_3) + a_3 m_{13} p_{13y} qdd_1 \sin(q_2 + q_3) + \\
& a_3 m_{13} p_{13z} qd_1^2 \cos(2q_2 + 2q_3) + m_{13} p_{13x} p_{13y} qdd_1 \sin(q_2 + q_3) + a_2 a_3 m_{13} qdd_2 \cos(q_3) + \\
& m_{13} p_{13x} p_{13z} qd_1^2 \cos(2q_2 + 2q_3) + a_3 m_{13} p_{13x} qd_1^2 \sin(2q_2 + 2q_3) + a_2 m_{13} p_{13x} qdd_2 \cos(q_3) - \\
& a_2 m_{13} p_{13z} qdd_2 \sin(q_3)
\end{aligned}$$

B.0.6 Grasp Force Estimation Raw Data

Table B.1: The measured and estimated grasp forces of the gripper.

LFG: Lateral Flexion Grasp				
Target Pose $q_2: 0.0rad$ $q_3: -1.5708rad$				
				
Measured mean grasping force (N) from loadcell	Commanded goal torques (Dynamixel)	Stall Torque Estimation (Nm)	$J_s(q)$: est. actuator torques	Actuator position (rad)
6.99 ± 0.36	50	0.38	$q_2: 0.32Nm$ $q_3: 0.11Nm$	$q_2: 0.6035 \pm 0.0070$ $q_3: -1.5364 \pm 0.0063$
12.31 ± 0.29	100	0.72	$q_2: 0.55Nm$ $q_3: 0.17Nm$	$q_2: 0.5546 \pm 0.0020$ $q_3: -1.5287 \pm 0.0079$
20.76 ± 0.16	150	1.06	$q_2: 0.9Nm$ $q_3: 0.25Nm$	$q_2: 0.5164 \pm 0.0006$ $q_3: -1.5331 \pm 0.0011$
29.07 ± 0.12	200	1.4	$q_2: 1.23Nm$ $q_3: 0.31Nm$	$q_2: 0.4807 \pm 0.0010$ $q_3: -1.5302 \pm 0.0012$
LEG: Lateral Extension Grasp				
Target Pose $q_2: 0.0rad$ $q_3: -0.6328rad$				
				
Measured mean grasping force (N) from loadcell	Commanded goal torques (Dynamixel)	Stall Torque Estimation (Nm)	$J_s(q)$: est. actuator torques	Actuator position (rad)
5.1 ± 0.05	50	0.38	$q_2: 0.35Nm$ $q_3: 0.17Nm$	$q_2: 0.1495 \pm 0.0031$ $q_3: -0.6113 \pm 0.0044$
8.8 ± 0.18	100	0.72	$q_2: 0.6Nm$ $q_3: 0.29Nm$	$q_2: 0.1029 \pm 0.0019$ $q_3: -0.6085 \pm 0.0006$
16.19 ± 0.03	150	1.06	$q_2: 1.08Nm$ $q_3: 0.5Nm$	$q_2: 0.0639 \pm 0.0007$ $q_3: -0.6076 \pm 0.0000$
17.25 ± 0.11	200	1.4	$q_2: 1.15Nm$ $q_3: 0.53Nm$	$q_2: 0.0587 \pm 0.0006$ $q_3: -0.6073 \pm 0.0007$

Appendix C



Gripper Development

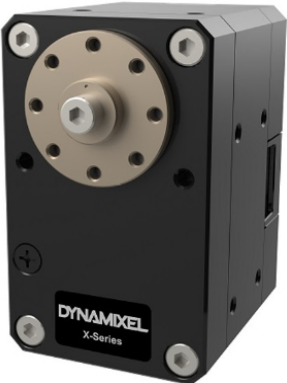
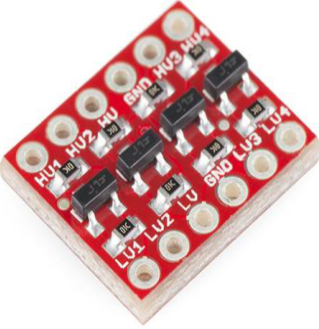

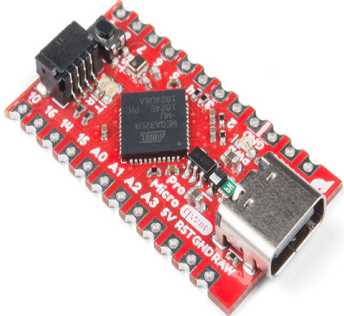
C.1 Acknowledgement of Collaborative Contributions

- Jamie Plowman (Lab Engineer) - Jamie provided assistance throughout the entire development cycle of the gripper between 2020 and 2021. His contributions included building the initial conceptual CAD model of the proposed gripper, providing advice to the candidate on circuit design, and offering counsel on sensor integration techniques.
- Jed Hodson (Undergraduate Student) - Jed dedicated his time to engineering groundwork in 2019 and 2020 through volunteering. His contribution involved working on the design, fabrication, electronics, and firmware of the first two iterations of the gripper, showcased in Figures 4.4(a) and 4.4(b). Jed's work laid the foundation for the engineering work that led to the development of the final research prototype, illustrated in Figure 4.4(c).
- Charles Raffaele (Undergraduate Student) - Charles volunteered his time developing robotic grasping demos that used trained computer vision models to detect clothing to grasp throughout 2021. He also created a ROS framework to control the Baxter robot and move the arms towards targeted grasping points.
- Chris Wu (Masters Student) - During 2021, Chris worked on optimising and training MaskRCNN models on custom data sets. These models were specifically designed to detect clothing. The computer vision modules were then utilised for the Baxter robot to see and grasp clothing in practical demonstrations.
- Dylan Morely (Undergraduate Student) - In 2022, Dylan contributed to the final research prototype (Figure 4.4(c)) by providing design work. His role involved collaborating with the Faculty of Arts and Design at the University of Canberra to create aluminium brackets for the gripper's link components, which formed grasps. He was also responsible for fabricating the main chassis components on the final gripper design.
- David Hinwood - David, the Ph.D. candidate of this thesis, was responsible for completing the entire project. He played a crucial role in designing the gripper and contributed to every project phase. David also worked


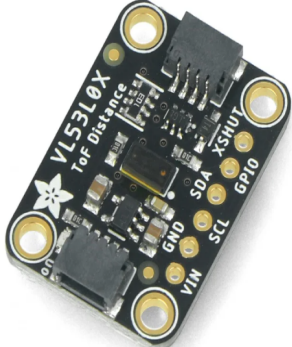
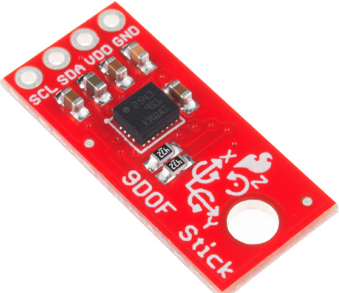

closely with all collaborators to design, fabricate, and program the electronics and firmware. Additionally, he was solely responsible for creating the final prototype's circuitry, electronics, and firmware. Furthermore, David was responsible for integrating the hand into a ROS framework, rigid-body modelling, system evaluation, and reinforcement learning research.

C.2 List of Electronic Components

Part Image	Description	Manufacturer
	<p>A4988 Stepper Driver - <i>POLOLU-1182</i> A breakout board that interfaces with the A4988 chip, this chip is a micro-stepping bipolar stepper motor driver. The circuit also contains a potentiometer to limit current flow and in-built temperature protections. This device uses between 8-35V and connects to the SparkFun Qwiic Pro microcontroller to operate the stepper motor.</p>	<p>Pololu</p>
	<p>NEMA 17 Stepper - <i>POLOLU-2267</i> The prismatic component of the gripper uses this stepper motor to actuate. The motor operates in steps, and each revolution has 200 steps. However, when interfaced with the A4988, micro-stepping for more refined motion is possible.</p>	<p>Pololu via SOYO</p>

Part Image	Description	Manufacturer
	<p>Dynamixel XM-430-350-R</p> <p>These rotational actuators make up the remaining three joints of the gripper. Each one operates a low-level impedance control appropriate for grasping applications. Controlling these actuators operates through a serial line interfacing to an RS-485 protocol (See the MAX485 breakout).</p>	Dynamixel
	<p>Logic level converter - BOB-12009</p> <p>These components are bi-directional logic level converters, enabling communication between chips that operate on differing circuit voltages. In this application, the logic level converters enable the Teensy device to communicate with the Qwiic Pro micro-controller and the MAX485 chip.</p>	Sparkfun
	<p>Teensy 4.0 - DEV-15583</p> <p>The Teensy 4.0 micro-controller is a high-speed processing unit commonly seen in robotic applications. In this application, the Teensy communicates between the host PC, the sensors, and the actuator controllers, acting as the primary system controller.</p>	PJRC
	<p>5Mhz Qwiic Pro - DEV-15795</p> <p>This controller acts as the primary interface to the stepper motor. By communicating the actuator operation details with the Teensy, this processor then controls the position and velocity behaviour of the prismatic actuator through the A4988 stepper driver.</p>	Sparkfun

Part Image	Description	Manufacturer
	<p>3V3 step down - <i>POLOLU-2857</i> This step-down voltage converter takes an input voltage from 4-36V and reduces it to 3.3V. It is capable of providing 2.6A of current.</p>	Pololu
	<p>5V step down - <i>POLOLU-2851</i> This step-down voltage converter takes an input voltage up to 38V and reduces it to 5V. It is capable of providing 5A of current.</p>	Pololu
	<p>12V step down - <i>POLOLU-2885</i> This step-down voltage converter takes an input voltage up to 40V and reduces it to 12V. It is capable of providing 15A of current.</p>	Pololu
	<p>MAX485 - <i>CE05154</i> This chip converts data from a serial communication line into the RS-485 protocol. This chip allows a pair of serial pins on the Teensy microcontroller to communicate with Dynamixel actuators.</p>	Core Electronics

Part Image	Description	Manufacturer
	<p>MPRLS Pressure Sensor - ADA-3965</p> <p>This breakout is pressure sensor which communicates barometric pressure data across the <i>i2c</i> line. The pressure measured comes from an enclosed silicon tube that acts as a bumper sensor to detect collisions. The tube attaches to central port of the breakout. The breakout uses the communicates the <i>i2c</i> interface to the Teensy controller.</p>	Adafruit
	<p>ToF Laser Sensor - ADA-3317</p> <p>This is a sensor that measures linear distance to a surface. Like the pressure sensor, this device also communicates to the Teensy with the <i>i2c</i> line. This sensor is used to calibrate the prismatic component by reading the rail displacement. However, it cannot be used in a closed loop form as the sensor takes approximately 100ms to take a single reading.</p>	Adafruit
	<p>9DoF sensing Stick - SEN-13944</p> <p>This breakout is a gyroscope device that uses the LSM9DS1, a motion-capture chip that contains a 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer. This sensor provides a robust motion sensing platform that communicates with the Teensy controller over the <i>i2c</i> interface.</p>	Sparkfun
	<p>CP2104 USB-to-Serial - POLOLU-1308</p> <p>This breakout carries a USB to UART bridge that enables a USB cable from a host PC to communicate with the Teensy. This device is integrated onto the circuit board such that the USB line will power the breakout while communicating with the gripper.</p>	Pololu

C.3 Microcontroller Communication

Table C.1: The available destination and commands for the serial protocol.

<i>DESTINATION</i> <i>uint8 integer</i>	<i>COMMAND</i> <i>uint8 integer</i>
JOINT_0	TARGET_POSITION TARGET_VELOCITY PRESENT_POSITION PRESENT_VELOCITY IS_MOVING ENABLED ERROR_STATUS
JOINT_1 JOINT_2 JOINT_3	TARGET_POSITION TARGET_CURRENT PRESENT_POSITION PRESENT_VELOCITY PRESENT_VOLTAGE PRESENT_TEMPERATURE PRESENT_LOAD ENABLED IS_MOVING ERROR_STATUS
SENSOR_IMU	REQUEST GYRO_ROLL GYRO_PITCH GYRO_ROLL
SENSOR_PRESSURE	PRESSURE_READING
TOF_LASER	REQUEST LASER_READING
EVAL	CONTROL_LOOP_DURATION DYNA_COMMAND_DURATION TEENSY_INTERFACE_DURATION QWIIC_INTERFACE_DURATION

C.4 Dynamic Parameters Estimation

The dynamics parameters in the modelling process differ slightly from the URDF values. This section of the appendix recalculates the inertia tensor and CoM parameters to align with the intended DH parameters. Two coordinate frames from the URDF do not require alteration when performing the modelling process. The inertia tensor for the prismatic actuator of the system simply defaults to zeros. The orientation of the EE frame is coincidentally in the correct orientation for both the URDF and modelling process. These aspects result in the inertia tensors from the URDF's first and last links as remaining equal for the DH parameters. For the remaining links, the inertia tensors about the CoM require reorientation. Reorienting the inertia tensor is a simple matter of taking a rotation matrix R_o constituting the desired reorientation sequence, with the inertia tensor I_o , and the reoriented inertia tensor can then be expressed as $R_o I_o R_o^T$. As URDFs differ fundamentally from the convention used in the modelling process, inertia tensors are initially estimated from the joint actuating the link. However, per

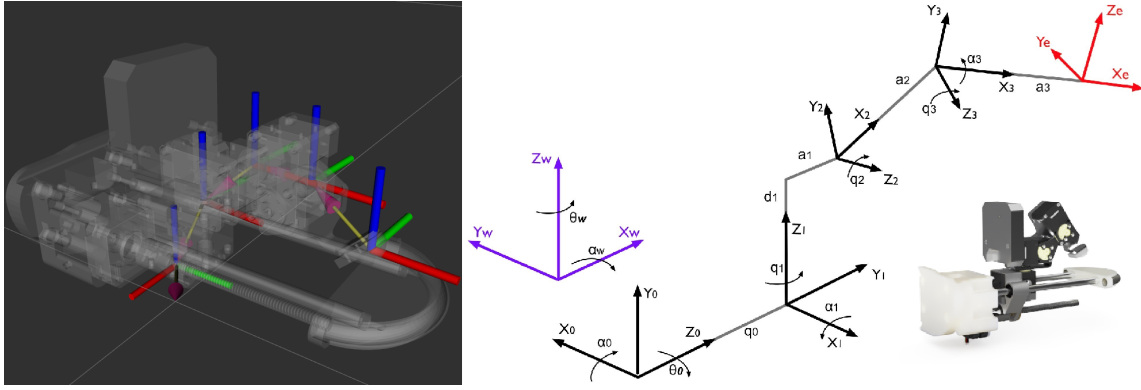


Figure C.1: A comparative view of URDF and DH coordinate frames.

the modelling process, inertia tensors about the CoM are taken from the orientation of the end of the DH row. The following equations demonstrate how the new inertia tensors about the CoM for both links 1 (Eq. C.1) and 2 (Eq. C.2) are calculated. Conveniently, for both inertia tensor reorientations, a simple rotation of $\frac{\pi}{2}$ is required.

$$I_{1urdf} = \begin{pmatrix} \frac{8005886927989945}{36893488147419103232} & \frac{1}{1000000} & \frac{308982963234635}{4611686018427387904} \\ \frac{1}{1000000} & \frac{2222832660882001}{9223372036854775808} & -\frac{1}{1000000} \\ \frac{308982963234635}{4611686018427387904} & -\frac{1}{1000000} & \frac{1724770570891843}{9223372036854775808} \end{pmatrix}$$

$$I_1 = R_x\left(\frac{\pi}{2}\right) \cdot I_{1urdf} \cdot R_x\left(\frac{\pi}{2}\right)^T \quad (C.1)$$

$$I_1 = \begin{pmatrix} \frac{8005886927989945}{36893488147419103232} & -\frac{308982963234635}{4611686018427387904} & \frac{1}{1000000} \\ -\frac{308982963234635}{4611686018427387904} & \frac{1724770570891843}{9223372036854775808} & \frac{1}{1000000} \\ \frac{1}{1000000} & \frac{1}{1000000} & \frac{2222832660882001}{9223372036854775808} \end{pmatrix}$$

$$I_{2urdf} = \begin{pmatrix} \frac{534955578137577}{18446744073709551616} & 0 & -\frac{1}{1000000} \\ 0 & \frac{3}{100000} & 0 \\ -\frac{1}{1000000} & 0 & \frac{6198106008766409}{295147905179352825856} \end{pmatrix}$$

$$I_2 = R_x\left(\frac{\pi}{2}\right) \cdot I_{2urdf} \cdot R_x\left(\frac{\pi}{2}\right)^T \quad (C.2)$$

$$I_2 = \begin{pmatrix} \frac{534955578137577}{18446744073709551616} & \frac{1}{1000000} & 0 \\ \frac{1}{1000000} & \frac{6198106008766409}{295147905179352825856} & 0 \\ 0 & 0 & \frac{3}{100000} \end{pmatrix}$$

Taking the noted discrepancies between the URDF and the modelling process, the CoM parameters are now recalculated for the modelling process. However some slight alterations are made when estimating the new CoM parameters. To start with, the URDF positions the actuators on the edge of the Dynamixel's bodies, despite the bracket coupling present on the final two joints. To that end, the modelling process assumes actuation takes place

from the centre of the Dynamixel's body for these joints. Therefore, one way of thinking about modifying the CoM value is that the CoM is being recalculated from to the next frame in the URDF and oriented to match the DH parameters. To that end, the transforms between joints alongside the URDF CoM values are used in the calculations. Additionally, the final coordinate frame in the URDF is positioned at the sensor. Whereas the dynamics parameters do not consider the sensor and simply include it's weight and distribution as part of the final link. Considering all the these details, the recalculation for each link's CoM to the modelling process is now present in Equations C.3, C.4, C.5 and C.6. Note, the function T denotes a homogeneous transformation matrix from a xyz vector or a specific translation along a axis with a subscript. The new CoM value for the modelling process of each link is denoted as $\text{CoM}_{i,\text{DH}}$.

For link 0, the CoM transform from the URDF's previous joint is shown below as $\text{CoM}_{0,\text{urdf}}$

$$\text{CoM}_{0,\text{urdf}} = \begin{bmatrix} 0.0 & 0.006 & 0.026 \end{bmatrix} (\text{xyz})$$

The transformation from the previous joint to the next connection point is expressed as J_{t0}

$$J_{t0} = \begin{bmatrix} 8.7e-05 & 0.022115 & 0.049537 \end{bmatrix} (\text{xyz}) \quad (\text{C.3})$$

Finally, the calculation of $\text{CoM}_{0,\text{DH}}$

$$\text{CoM}_{0,\text{DH}} = \left(T(\text{CoM}_{0,\text{urdf}})^{-1} \cdot T(J_{t0}) \right)^{-1}$$

$$\text{CoM}_{0,\text{DH}} = \begin{bmatrix} -0.0001 & -0.0161 & -0.0235 \end{bmatrix}$$

The following equations use the same denominations as those given in Eq. C.3.

$$\text{CoM}_{1,\text{urdf}} = \begin{bmatrix} -0.013 & 0.0 & 0.034 \end{bmatrix} (\text{xyz})$$

$$J_{t1} = \begin{bmatrix} 0.019959 & 0.019105 & 0.020579 \end{bmatrix} (\text{xyz})$$

In the calculation of $\text{CoM}_{1,\text{DH}}$, an additional translation along the z-axis accounts for the bearing attachment mechanism (remembering that the CoM for this link with rely on the the third actuator's position)

$$\text{CoM}_{1,\text{DH}} = \left(T(\text{CoM}_{1,\text{urdf}})^{-1} \cdot T(J_{t1}) \cdot R_x\left(\frac{\pi}{2}\right) \cdot T_z(0.019105) \right)^{-1}$$

$$\text{CoM}_{1,\text{DH}} = \begin{bmatrix} -0.0330 & 0.0134 & 0 \end{bmatrix}$$

(C.4)

$$\begin{aligned}
\text{CoM}_{2,\text{urdf}} &= \begin{bmatrix} 0.034 & -0.019 & 0.011 \end{bmatrix} (\text{xyz}) \\
J_{t2} &= \begin{bmatrix} 0.035744 & 0.0011 & -0.000297 \end{bmatrix} (\text{xyz}) \\
\text{CoM}_{2,\text{DH}} &= \left(T(\text{CoM}_{2,\text{urdf}})^{-1} \cdot T(J_{t2}) \cdot R_x\left(\frac{\pi}{2}\right) \cdot T_z(0.019105) \right)^{-1} \\
\text{CoM}_{2,\text{DH}} &= \begin{bmatrix} -0.0017 & 0.0113 & 0.0010 \end{bmatrix}
\end{aligned} \tag{C.5}$$

$$\begin{aligned}
\text{CoM}_{3,\text{urdf}} &= \begin{bmatrix} 0.022 & -0.019 & -0.007 \end{bmatrix} (\text{xyz}) \\
J_{t3} &= \begin{bmatrix} 0.044491 & -0.019002 & -0.013363 \end{bmatrix} (\text{xyz})
\end{aligned}$$

An additional transform is present to account for the position of the sensor, see Section 4.2

and the variable of Δ_z

$$\begin{aligned}
\text{CoM}_{3,\text{DH}} &= \left(T(\text{CoM}_{3,\text{urdf}})^{-1} \cdot T(J_{t3}) \cdot T_z(0.013363) \right)^{-1} \\
\text{CoM}_{3,\text{DH}} &= \begin{bmatrix} -0.0225 & 0 & -0.007 \end{bmatrix}
\end{aligned} \tag{C.6}$$

Appendix D

Reinforcement Learning Supplemental Materials

D.1 Arbitrary Grasping Environment Details

D.1.1 Learning Vector Details

State space attribute (24 elements)	Description	Normalization Range (-0.75 — 0.75)	
		Lower Bound	Upper Bound
<i>Grey cells indicate action space values</i>	<i>Yellow cells indicate components that are subject to noise from domain randomization</i>		
SENSOR_TCP_X	The x position of the sensor calculated from Equation x.s using the present position values.	-0.05	0.25
SENSOR_TCP_Y	The y position of the sensor, calculated from Equation y.s using the present position values.	-0.15	0.15
SENSOR_TCP_Z	The z position of the sensor, calculated from Equation z.s using the present position values.	-0.1	0.15
SENSOR_TCP_P	The pitch orientation aspect of the sensor, calculated from Equation p.s using the present position values.	-1	2.2
J0_TARGET_POSITION	The target position given to the prismatic actuator in meters.	0.0	0.08
J1_TARGET_POSITION	The target position given to the first rotational actuator in radians.	0.872665	2.26893
J2_TARGET_POSITION	The target position given to the second rotational actuator in radians.	0.0	1.5708
J3_TARGET_POSITION	The target position given to the third rotational actuator in radians.	-1.5708	0.174533
J0_PRESENT_POSITION	The present position of the prismatic rail component in meters.	0.0	0.08

J1_PRESENT_POSITION	The present position of the first rotational actuator in radians.	0.872665	2.26893
J2_PRESENT_POSITION	The present position of the second rotational actuator in radians.	0.0	1.5708
J3_PRESENT_POSITION	The present position of the third rotational actuator in radians.	-1.5708	0.174533
J0_TARGET_VELOCITY	The target velocity command given to the prismatic rail (units in meters/sec).	-0.055	0.055
J1_TARGET_CURRENT	The target current value given to the first rotational actuator (units in Dynamixel Current steps, see cite).	15	50
J2_TARGET_CURRENT	The target current value given to the second rotational actuator (units in Dynamixel Current steps, see cite).	15	50
J3_TARGET_CURRENT	The target current value given to the third rotational actuator (units in Dynamixel Current steps, see cite).	15	50
J0_PRESENT_VELOCITY	The present velocity of the prismatic actuator (units in meters/sec).	-0.055	0.055
J1_PRESENT_VELOCITY	The present velocity of the first rotational actuator (units in rad/sec).	-4.81711	4.81711
J2_PRESENT_VELOCITY	The present velocity of the second rotational actuator (units in rad/sec).	-4.81711	4.81711
J3_PRESENT_VELOCITY	The present velocity of the third rotational actuator (units in rad/sec).	-4.81711	4.81711
FORCE_Z	The force reading of the sensor in the Z direction (units in Newtons)	-5	15
GYRO_X	The normalized reading of the gyroscope about the x-axis (units in radians)	-pi	pi
GYRO_Y	The normalized reading of the gyroscope about the y-axis (units in radians)	-pi	pi
EC_DISTANCE	The distance traversed across the environment while grasping. Given by the reward function which tracks this parameter (units in meters)	-0.1	0.1

D.1.2 Domain Randomisation Components

Simulation Components			Domain Randomisation Configurations		
<i>Element of simulation</i>	<i>Details</i>	<i>Units</i>	<i>Lower Bound</i>	<i>Upper Bound</i>	<i>Noise Type</i>
Gripper Orientation (ZYX)	The roll-pitch-yaw angles of the gripper's orientation relative to the world coordinate frame w. Noise is added to each ground truth value read from the simulation.	Radians	-0.0523599	0.0523599	Uniform
Rotational Joint Positions	The position of the rotational joints. Noise is added to each ground truth value read from the simulation.	Radians	-0.05	0.05	Uniform
Rotational Joint Velocities	The velocities of the rotational joints. Noise is added to each ground truth value read from the simulation.	Radians/sec	-0.03	0.03	Uniform

Simulation Components		Domain Randomisation Configurations			
Rotational Joint Position Gain	The position gain for the underlying PD controller that operates the simulated rotational joints. This value is randomly generated between the noise bounds each time the simulator performs a hard reset.	N/A (float constant)	0.25	0.44	Uniform
Rotational Joint Stall Torque Equation	The linear expression that determines the stall torque of the actuators. The default gradient from Eq. X remains constant while offset value varies between the boundaries	N/A (float constants)	0.01	0.08	Uniform
Prismatic Joint Position	The position of the prismatic rail mechanism. Noise is added to each ground truth value read from the simulation.	Meters	-0.002	0.002	Uniform
Prismatic Joint Velocity	The velocity of the prismatic rail mechanism. Noise is added to each ground truth value read from the simulation.	Meters/sec	-0.002	0.002	Uniform
Environment Gravity	The gravity force of the simulation. hardreset.	Newton (N)	-9.1233	-10.4967	Uniform
Link mass (default value of m)	The mass of each link in kilograms. The default value is held in the URDF. hardreset.	Kg	m*0.8	m*1.2	Uniform
Orientation of the environment surface	The orientation of the environment surface about the x-axis and y-axis. hardreset	Radians	-0.0523599	0.0523599	Uniform
Texture of the environment surface	The are four URDF files creating various environment surfaces for the gripper. Upon every simulator hard reset, the next surface in the sequence is embedded into the simulation.	N/A (integer constant)	0	3	Sequential
F_z value read from the fingertip force sensor.	A small amount of random noise applied to the force sensor's F_z readings.	Newton (N)	-0.35	0.35	Uniform
Exponential function defining degradation of F_z readings.	An exponential function defined as $f(x) = c \cdot a^{kx} + d$ where x is the input displacement (in mm) from the sensors centre point of Figure 5.7. The values of c , a , k and d are chosen from 16 user-defined exponential functions.	N/A	N/A	N/A	Uniform
Friction of the environment surface	The lateral friction parameter for the environment surface. (Add footnote)	N/A (float constant)	0.02	1.25	Uniform
Friction of the gripper's sensor	The friction of the triaxial force sensor.	N/A (float constant)	0.65	1.0	Uniform

D.2 Reward Class Pseudocode

Algorithm 4 The class structure hosting the reward function to encourage *EC* grasping

Input: (rs) \triangleright Requires an input reward scaling value upon class initialisation.
 $EC_{dist} \leftarrow 0$ \triangleright Set the distance travelled while traversing the environment to 0.
 $R_{acc} \leftarrow 0$ \triangleright This variable tracks the accumulated reward while dragging along the environment.
 $R_{sg} \leftarrow 100 \times rs$ \triangleright Initialises the reward value for a successful grasp.
 $R_{ic} \leftarrow 25 \times rs$ \triangleright Initialises the reward value for initially colliding with the environment.
 $R_{ec} \leftarrow 1500 \times rs$ \triangleright Initialises the reward value for traversing along the environment.
function CALCULATE_REWARD(s, s') \triangleright Inputs include the previous state s and present state s' .
 bool $coll_s \leftarrow$ if s indicates the gripper was previously in contact with the environment
 bool $coll_{s'} \leftarrow$ if s' indicates the gripper is currently in contact with environment
 bool $grasp_{s'} \leftarrow$ if s' indicates the gripper is currently in a grasp pose
 if $coll_{s'}$ **then**
 if $coll_s$ **then**
 Calculate the change in euclidean distance of the fingertip from the static plate
 $ec_\delta = \text{CALCULATE_EUCLIDEAN_DISTANCE_TRAVERSED}(s, s')$ $\triangleright ec_\delta$ in meters
 $EC_{dist} += ec_\delta$ $\triangleright ec_\delta$ is negative if moving away from the static plate.
 if Fingertip moves further away from the plate **then** \triangleright See **Note 1**.
 return $-R_{ec} \times ec_\delta$ \triangleright Penalise a traversing motion moving away from the plate.
 else if Fingertip moves closer to the plate **then** \triangleright See **Note 1**.
 $R_{acc} += R_{ec} \times ec_\delta$ \triangleright Add the reward from distance travelled to R_{acc} .
 return $R_{ec} \times ec_\delta$ \triangleright Reward a traversing motion moving towards the plate.
 end if
 else if not $coll_s$ **then**
 return R_{ic} \triangleright Return the reward value for making initial contact with the environment.
 end if
 else if $grasp_{s'}$ and $EC_{dist} > 0.0015$ **then** \triangleright Conditions for a successful grasp (units in meters).
 return R_{sg} \triangleright Return the reward for a successful grasp attempt
 else if The policy abandons the grasping trajectory **then** \triangleright See **Note 2**.
 $rw \leftarrow -R_{acc}$ \triangleright Set the returned reward to the accumulated reward value R_{acc} .
 $EC_{dist} \leftarrow 0$ \triangleright Reset the distance travelled to 0 for a new grasping attempt.
 $R_{acc} \leftarrow 0$ \triangleright Reset the accumulated dragging reward to 0 for a new grasping attempt.
 return rw \triangleright Return a penalising value based on the reward received while dragging.
 end if
 return 0 \triangleright Return a reward of 0 if no previous conditions are triggered.
end function

Notes:

1. Determining whether the fingertip moved closer or further away from the plate uses Equations 4.12-4.15 and the known position of h described in section 5.2.2.
 2. Determining whether abandonment of a trajectory or grasping motion has occurred was a series of conditional Boolean evaluations based on kinematic configurations. An example occurs when the gripper removes the fingertip from the environment while moving away from the static plate. Alternatively, this condition activates if the gripper moves into a grasping configuration without appropriately traversing along the environment surface first. When this abandonment occurs, the reward function returns a penalised value equivalent to the sum of positive rewards received while dragging along the environment.
-