



Trinity College Dublin

Coláiste na Tríonóide, Baile Átha Cliath

The University of Dublin

SCHOOL OF ENGINEERING

Grasping Moving Objects: Adaptive Motion Through Tactile Sensing

PATRICK LYNCH

OCTOBER 4, 2022

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF

DOCTOR IN PHILOSOPHY IN SCHOOL OF ENGINEERING

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish.

Signed: Patrick Lynch

Date: 06/04/22

Summary

To operate in unstructured settings, robots require control strategies that effectively respond to temporal changes in their environment. Grasping a moving object greatly increases the demands placed on the robot's sensory, computation and actuation systems. There has been a tendency in previous research to use control strategies that require the robot to achieve a near-perfect interception of the object. This has typically been achieved within controlled laboratory settings that have access to advanced motion-tracking exocentric sensing, powerful computational systems, and highly-accurate robotic actuators. The same accuracy is considerably more difficult to achieve in a real-world application. Additionally, there is a gap in the existing research literature on gripper control strategies for grasping moving objects. Where grasping moving objects has been studied previously, it has been common for researchers to simplify the grasping problem by replacing the gripper with a net or basket.

The aim of this project was to explore how adaptive grasping motions, informed by real-time tactile feedback, can improve grasp robustness when grasping moving objects. It was hypothesised that an adaptive grasping motion, based on real-time tactile sensor feedback, could mitigate the effects of errors in the interception of a ball by a robotic gripper, resulting in a more robust control strategy for grasping moving objects.

First, an experimental methodology was developed to enable a systematic examination of a gripper's ability to grasp a moving object under a range of grasping conditions, including: the grasp initiation time, the initial position of gripper-object contact, and the object speed. The procedure involved a two-finger gripper tasked with grasping a ball rolling toward it on a horizontal plane using a power grasp. Simplifying the grasping problem in this way, using a two-dimensional task and eliminating the effect of object orientations, ensured a high level of repeatability and enabled individual aspects of the grasping conditions to be examined in isolation.

A virtual environment was developed to test the performance of a two-finger gripper grasp-

ing a moving object in simulation. The performance of two grasping strategies were evaluated. The first strategy was a proxy to existing approaches that rely solely on an estimation of a position and time at which the gripper can attempt to grasp the ball. The second strategy was a reactive control strategy, which initiated the grasp when the ball made contact with the fingers, monitored the position of the ball during the grasp, and adapted the gripper's motion accordingly. This adaptive motion was implemented by creating a set of simple heuristics which determined how to act based on tactile sensing. Testing revealed that the reactive grasping strategy outperformed its predictive counterpart, achieving an average grasping success rate of 58% compared to 36% across the same range of grasping conditions.

Next, to verify that the results from simulation translate to the real-world, a physical experimental apparatus was created and tests were performed using the same predictive and reactive strategies. During experiments with the physical testing rig, the reactive control strategy achieves a grasping success rate of 80% while the predictive controller achieved a success rate of 71%. This dropped to 70% and 51% for grasp timings which were offset from the optimum by just 5ms and 10ms respectively. Results of both simulated and real-world testing demonstrate the additional grasp robustness achieved by an adaptive grasping motion, informed by real-time sensor data. Despite the observed performance improvement, the heuristic-based implementation of the reactive control strategy was observed to possess inherent limitations.

Finally, to optimise the performance and overcome these limitations, an additional reactive control strategy, which relied on a neural network agent, was developed. The agent was trained, first using supervised learning, then in simulation using reinforcement learning. The model was deployed on the real-world gripper and its performance quantified in accordance with the same methods used in previous experiments. Results show that, of the two proposed, reactive strategies, the neural network based strategy, utilising a learned adaptive motion, outperformed its heuristic-based counterpart. It achieved an average grasping success rate of 90% across the same range of grasping conditions.

Reactive grasping motions, informed by tactile sensing, were shown to improve grasp robustness to inaccurate object interceptions. This thesis presents foundational research regarding this type of novel, adaptive grasping strategy demonstrating a significant improvement to a robot's ability to grasp moving objects in unstructured environments and enables exciting new applications in this field.

Publications

The following manuscripts have been published during the course of this PhD:

- “Adaptive Grasping of Moving Objects through Tactile Sensing”, Patrick Lynch, Michael F. Cullinan, Conor McGinn, *Sensors*, 2021, doi: 10.3390/s21248339
- “Using Tactile Sensing to Improve Performance when Grasping Moving Objects”, Patrick Lynch, Conor McGinn, *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2021, doi: 10.1109/SMC52423.2021.9658609
- “Presenting Botmark: a Computer Benchmark for Service Robotics”, Andrew Murtagh, Patrick Lynch, Conor McGinn, *IEEE International Conference on Robotic Computing (IRC)*, 2020, doi: 10.1109/IRC.2020.00046
- “Meet Stevie: a Socially Assistive Robot Developed Through Application of a ‘Design-Thinking’ Approach”, Conor McGinn, Eamonn Bourke, Andrew Murtagh, Cian Donovan, Patrick Lynch, Michael F. Cullinan, Kevin Kelly, *Journal of Intelligent & Robotic Systems*, 2020, doi: 10.1007/s10846-019-01051-9

The following manuscripts have been submitted and are awaiting review:

- “Improved Grasping of Moving Objects through Learned Adaptive Behaviour”, Patrick Lynch, Michael F. Cullinan, Conor McGinn, *Transactions on Systems, Man, and Cybernetics : Systems*, 2022

Acknowledgements

The work outlined in this thesis could not have been done without a support system full of generous and kind people.

Firstly, to my family. You have given me the best possible start in life, encouraged me to chase every opportunity and supported me every time. Without doubt, I would not be where I am today without you. To my Mum, your patience and unwavering dedication to giving me and Rosemary every possible opportunity is astounding. The sacrifices you have made to give up every evening and weekend to take us to every sports club, study session, and music class in a fifty kilometer radius will never be forgotten. To my Dad, you are the reason I am an engineer. The skills I learnt by watching, and occasionally helping, you to make gates, and fix bailers contributed more than you know to the work in this thesis. Finally to my little sister Rosemary, thank you for your support, your kindness and for your constant companionship throughout my entire life (give or take 18 months), for you there is only one question remaining "Wanna fight?".

Next to my girlfriend, Shannon. Your dedication is incredible and your passion inspiring. If everyone put as much effort into, and took as much pride from their work, as you do the world would be a much better place. Thank you for your unbelievable support throughout this entire process. The thesis is the result of countless missed date nights, skipped weekends and holidays which were never booked. Your patience, encouragement and love though it all was essential. You always knew when the deadlines were closing in around me and you always knew what to do, I could not have done any of this without your support.

RAIL is an energetic, innovative and welcoming community of people within the Parsons building in Trinity College Dublin. It can honestly be said help will always be given in RAIL, to those who ask for it. So many people have come and gone during my time in RAIL and by adding to the community, each contributed in some small, indirect way to this project. That said, there are a few people whose contribution is neither small nor

indirect. These are some of the best and most talented people I have ever had the privilege to work with. First, Prof. Kevin Kelly, you have been a great advocate, and a great mentor. Your work has improved the educational experience of so many, including almost everyone I talk about below this point. I am so grateful to be a part of what you have created. Eamonn Bourke, your talents span from design to cocktails. Legend has it that your eyeball can measure the turning circle of a robot two rooms over, unless it has suspension, in which case it can't exist because robot's don't have suspension. Your consultancy on this and all future projects is greatly appreciated. Robbie Fryers, the man of obscure references and ridiculous words, you're a true gem to anyone who calls you a friend. Talita, the world needs at least another 30 of you. You're truly a woman on a mission and an inspiration to everyone around you. Thank you for inspiring me. Vanessa, my comrade in arms, or at least whatever the PhD equivalent of arms is, thank you for joining me in the misery of doing a PhD. It truly was incredible to have someone struggling with the same challenges, who I could share the experience with. I hope you are excelling in Brazil. Andrew Murtagh, a co author and a friend. You've listened to endless rambles of my C++ bugs and my rants about other stuff, and I thank you for that. Moyin and Adam, my fellow robotic PhD hopefuls, the pandemic has been tough for our small, world renowned research group and it has often felt like we three, were all that was left. With your help I hope that we can become the foundations of something great.

Aside from RAIL there is another group of people who's contribution to this thesis has been equally important. Akara, currently a small start up but which is perfectly posed for world domination in about two years, tops. I need to thank all of you, not just for letting me use your office and claim the position of office manager, but for your friendship, support, kindness, and constant stream of great coffee. By the way Niamh, I still think we should both quit and start a true crime podcast.

Despite the enormous contribution of the people in both the RAIL and Akara communities, there is undoubtedly one person who stands out above all others. To Michael, I am eternally grateful for all that you have done, there is no way this project would have reached its completion without your support. You entirely don't understand

the impact you have on those who have the privilege of working with you. Your knowledge and skill is out weighted only by your complete selflessness when there is an opportunity to help someone else. On the downside, I'm pretty sure we are both enablers of the other work addiction which is entirely out of control. I look forward to less late nights and less weekends in the office.

Finally, my supervisor, Dr. Conor McGinn. Your passion for robotics is unquestionable and it inspires all those around you. You are the reason this project was even possible, and that students are currently studying and researching robotics in Trinity, a truly remarkable accomplishment. Thank you for creating the environment which I have had the privilege of been a part of for almost 7 years. Thank you for your time, your efforts, and your belief in me. I wish you the best, I have no doubt Akara is destined for greatness.

Contents

1	Introduction	1
1.1	Motivation and Challenges	2
1.2	Research Objectives	3
1.3	Research scope	5
1.4	Contributions	5
1.5	Overview	7
2	Literature Review	9
2.1	Interacting with a dynamic object	9
2.1.1	Vision sensing detects and localises the object	10
2.1.2	Modeling and estimating object trajectory	10
2.1.3	Identifying an appropriate interception point	11
2.1.4	Moving the end-effector to the interception point	12
2.1.5	Intercepting the object	12
2.2	Existing robotic grippers	14
2.2.1	Underactuated robotic grippers	20
2.3	Sensing for robotic grippers	22
2.3.1	Joint angle sensing	22
2.3.2	Visual sensing in robotic grasping	24
2.3.3	Tactile sensing in robotic grasping	26
2.3.4	Visuotactile sensing in robotic grasping	32
2.4	Examples of interactions with dynamic objects	32

2.4.1	Examples where the object is on a conveyor belt	33
2.4.2	Hitting or batting an object	35
2.4.3	Capturing a ball	35
2.4.4	Grasping a ball	36
2.4.5	Grasping an irregular object	36
2.4.6	Limitations of current approaches	37
2.5	Machine learning techniques in robotics	38
2.5.1	Models and agents	39
2.5.2	Learning	42
2.6	Robotic grasping in simulation	47
2.7	Summary	50
3	Methods	53
3.1	Experimental methodology	53
3.1.1	Gripper morphology chosen for testing	55
3.1.2	Choice of test object	56
3.1.3	Positional interception error	57
3.1.4	Timing of grasp initiation	58
3.2	Velocity of the test object	59
3.3	Statistical analysis	60
3.4	Strategies for grasping moving objects	61
3.4.1	Predictive grasping strategy	62
3.4.2	Reactive grasping strategy	63
3.5	Summary	66
4	Validation through Simulation	69
4.1	Simulator Development	69
4.1.1	Gripper embodiment in simulation	71
4.1.2	Simulation of the underactuated mechanism	74
4.1.3	Control strategies and implementation	76

4.1.4	Controlling independent variables	78
4.1.5	Demonstration of simulated sample grasp	79
4.2	Calibration of the simulation	80
4.3	Determining the optimum grasp timing	83
4.4	Experimental Procedure	85
4.5	Results	85
4.6	Discussion	90
5	Validation through Real World Testing	95
5.1	Development of a real-world gripper	95
5.1.1	Requirements of the real-world gripper	96
5.1.2	Real-world gripper geometry	98
5.1.3	Design of the actuation systems for the real-world gripper	100
5.1.4	Design of the sensing systems for the real-world gripper	103
5.1.5	Selection of the computational systems for the real-world gripper	108
5.1.6	Control strategies and implementation	109
5.1.7	High gripper robustness	111
5.1.8	Gripper modularity and reconfigurability	111
5.1.9	Gripper cost	112
5.2	Development of a real-world testing apparatus	113
5.2.1	Choice of test object	113
5.2.2	Controlling object velocity	113
5.2.3	Controlling the position of initial gripper-object contact	114
5.2.4	Controlling the grasp initiation time	115
5.3	Summary of real-world gripper and apparatus design	116
5.4	Determining the optimum grasp timing	119
5.5	Experimental procedure	120
5.6	Results	122
5.7	Discussion	126

6	Improvement through Machine Learning Techniques	131
6.1	Machine learning agent	132
6.1.1	Choosing a neural network	132
6.1.2	Structure of the network used	133
6.2	Training the agent	138
6.2.1	Supervised learning	138
6.2.2	Reinforcement learning	140
6.2.3	Adaption for use in the real-world	143
6.3	Deploying the neural network control strategy on the ESP	143
6.3.1	ML control strategy implementation	144
6.4	Experimental procedure	144
6.5	Results	145
6.6	Discussion	150
7	Conclusion	153
7.1	Reflection on the contributions of this thesis	153
7.2	Applicability of this research	157
7.3	Limitations and future work	158
A1	Appendix	187
A1.1	Manufacture of custom tactile sensors	187

List of Figures

All figures presented in this thesis are available at <https://drive.google.com/drive/folders/1TXKr90VbsHUsGL9070wnWGsheWHhE-yc?usp=sharing>

1.1	Flow chart outlining the stages taken by a robot autonomously interacting with a dynamic object	5
2.1	Illustration of Rollin Justin predicting the future trajectory of two balls simultaneously.	11
2.2	Sample of existing low prehensor grippers, (a) Robotiq 2F-85 (b) OnRobot RG2 (c) Kuka youBot platform	15
2.3	Examples of existing 3 fingered robotic grippers (a) Schunk SDH (b) Barrett Hand (c) Robotiq 3-finger adaptive gripper	16
2.4	Sample of existing anthropomorphic robotic grippers. (a) DLR-Hand-II (b) Allegro Hand (c) Shadow Hand (d) Schunk SVH (e) TWENDY ONE (f) iCub (g) James	18
2.5	Examples of existing soft robotic grippers. (a) Particle jamming gripper (b) OctoGripper (c) RBO Hand 2	19
2.6	Example of an existing commercially available vacuum gripper	20
2.7	uSkin curved product offered by Xela Robotics, specifically designed to be used as the fingertip for the Allegro robot hand. (a) shows a rendering of the sensor and (b) shows a photo of uSkin deployed on the gripper	28

2.8	TakkTile pressure based tactile sensors by the Soft Robotics Toolkit, (a) shows the sensor itself, (b) the sensor implemented on the ReFlex robotic gripper	30
2.9	Sample of existing research of autonomous interaction between robot and dynamic object, mapped based on object complexity and the nature of the environment	34
2.10	Illustration of the difference between (a) shallow neural networks (SNN) and (b) deep neural networks (DNN)	41
2.11	Large scale data collection setup used to create the Google Grasp Dataset. 14 robots collected 800,000 grasp attempts	47
2.12	A simulated and real-world Baxter robot learning to grasp using reinforcement learning. The simulation is conducted using the Gazebo robotic simulator	48
2.13	Robotics simulator evaluating iCub's ability to grasp a moving object	50
3.1	Offset between gripper-object contact point and gripper centre, representing positional error in the interception of a moving object by a gripper	55
3.2	Diagram of gripper morphology, including labels which are used to refer to the different fingers, phalanges and joints	56
3.3	Illustration of the positional offset, (a) no induced positional offset, (b) a small induced positional offset (c) a large induced positional offset	58
3.4	Illustration of the angle of incidence, which is constant at 90° for each test	60
3.5	Flow chart illustrating the predictive strategy	63
3.6	Flow chart illustrating the reactive strategy	65
3.7	Illustration of the reactive grasping strategy: (a) Stage 1: Ball approaches gripper with an induced spatial offset (b) Stage 2: Ball makes contact with the gripper, a tactile sensor in the right finger triggers the grasp (c) Stage 3: Gripper begins to grasp the ball, this involves moving laterally to the right, and closing the left finger (d) Stage 4: Grasp is completed by closing the right finger	66

4.1	Diagram of the simulated gripper, including labels which are used to refer to the different fingers, phalanges and joints	72
4.2	Illustration of the link hieracharcy described in the sdf file.	72
4.3	Drawing of the simulated gripper showing the critical dimensions. All dimensions are in meters	73
4.4	Screenshot of the simulated gripper	74
4.5	Illustration of the sequential closing motion which results from the use of the underactuated mechanism	75
4.6	Illustration of the final position of the gripper after grasping a ball. The grasp is asymmetrical since the position of the ball is offset from the center of the gripper and the underactuated mechanism adapts the final grasp configuration accordingly	76
4.7	Graphs showing data collected during an example of a successful grasp using the simulated gripper. (a) Shows the ball and gripper position and (b) shows the joint angles, during a successful grasp.	79
4.8	Joint angles recorded during calibration grasps (a) Real-world grasp of small object (b) Simulated grasp of small object (c) Real-world grasp of test object (d) Simulated grasp of test object (e) Real-world grasp of large object (f) Simulated grasp of large object	82
4.9	Data used to find the optimum time to initiate the grasp, t_{Opt} . (a) Graph showing grasping success rate relative to the grasp initiation time (b) Graph showing grasping success rate relative to the spatial offset for multiple different grasp initiation times	84
4.10	Diagram of the simulated gripper including a visualisation of the tested object speeds and object-gripper contact positions.	85
4.11	Results for different speeds of the moving object: (a) 804mm/s, (b) 895mm/s, (c) 987mm/s, and (d) 1078mm/s	87
5.1	Bespoke gripper developed for real-world testing, represented as (a) a photo of the gripper and (b) a drawing of the gripper	96

5.2	Terminology used to refer to parts of the gripper	98
5.3	Drawings of the final gripper design, showing key dimensions of (a) the 4 unique parts used to make up the gripper, and (b) the assembled gripper. All dimensions are in millimeters.	99
5.4	Breakdown of the moments experienced by the gripper during grasping, categorised by (a) clockwise moments and (b) anticlockwise moments . .	101
5.5	Illustration of the gripper at various stages throughout the closing motion, showing the sequential closing motion	102
5.6	Visualisation of the lateral motion of the gripper	103
5.7	Joint sensing used on the bespoke gripper, (a) The potentiometer and (b) the implementation of the potentiometer on gripper	104
5.8	A demonstration of the advantage of implementing an air gap in the tactile sensors. This increases sensitivity while reducing cross-talk between different axis. (a) Without an air gap (b) Incomprehensibility causes unpredictable material flow (c) Introduction of an air gap (d) Displacement is in direction of the contact force	107
5.9	The Hall effect sensor used to obtain the tactile feedback was the MLX90393. This shows the schematic of the PCB used.	107
5.10	The computational unit chosen for the bespoke gripper was the ESP-32 microcontroller, specifically the Node MCU ESP-Wroom-32 Development Module	109
5.11	Renderings of the finger design. (a) Assembled Finger, View 1 (b) Assembled Finger, View 2 (c) Assembled Finger, View 3	111
5.12	Renderings of the finger design. (a) Exploded View (b) Exploded view of 1 side of the finger (c) 4 unique parts required.	112
5.13	Ball trajectory is controlled by an inclined track. A U-channel aluminium extrude (shown in cross-section) is used as the track.	114

5.14	Experimental apparatus focused on how positional errors in the object interception are introduced. (a) No error , (b) Small induced error (c) Large induced error	115
5.15	Experimental Rig, focused on the light gate (a) Illustration of the light gate, (b) Illustration showing the ball breaking the light gate (c) Photograph of the light gate on the real world apparatus	116
5.16	Experimental apparatus, (a) Photograph of the experimental apparatus from above, (b) An elevation view diagram of the experimental apparatus (c) A plan view diagram	118
5.17	Drawing of the final gripper design from three perspectives. Plan, elevation and end view	119
5.18	Illustration of the experimental procedure. The ball is set in motion by the release of a servo arm (1). The ball rolls down the incline (2) and through the light gates which records its velocity and notifies the control strategy (3). The gripper grasps (or attempts to grasp) the ball and the trapdoor is released to determine if the full mass of the ball can be supported by the gripper (4)	121
5.19	Tests were performed at 3 ball speeds and 5 spatial offsets as indicated in this graph.	121
5.20	Grasping Success Rate Vs Spatial Offset, Averaged Across all Test Speeds .	123
5.21	Successful grasp rates, observed over 10 tests under each set of conditions. (a) Predictive control strategy with 0ms temporal offset (b) Predictive control strategy with 5ms temporal offset (c) Predictive control strategy with 10ms temporal offset (d) Reactive control strategy	123
6.1	Illustration of the structure of the neural network used. 'I' represents the input nodes, 'O' the output nodes. 'n' represents the nodes in the hidden layers with the superscript showing the layer it belongs to and the subscript the node number within that layer.	135
6.2	Illustration of the output nodes of the neural network used	135

6.3	Illustration of the input nodes of the neural network used	137
6.4	Grasp showing the progress of the supervised pretraining, graphing errors over training iterations. (a) shows the errors in the network output associated with stepsize of the lateral motion position controller, (b) shows the errors in the network output associated with lateral movement limit	139
6.5	Plots which visualise the performance at difference stages during the training. (a) shows the performance of the pretrained network before RL is used, (b) shows the performance of the same network after RL is used to train it.	142
6.6	Performance of the the Neural Network Strategy at a range of object speeds and gripper-object contact points	146
6.7	Comparison of the performance of a Neural Network strategy with the performance of previously tested strategies, outlined in Chapter 5. The data presented is an average across all test speeds. Furthermore, each data point presented for the neural network's performance is an average of the positive spatial offset with its negative counterpart.	146
A1.1	3D Rendering of the molding process, showing (a) an exploded view (b) the drillbit in place (c) the magnet in place (d) the silicon in place with the lid off (e) the mold closed, and finally (f) the mold closed and secured with bolts	187
A1.2	Images of the 3D printed mold and molding process. (a) mold view 1 (b) mold view 2 (c) mold view 3 (d) mold with drillbit in place (e) mold with magnet in place (f) closed mold (g) resulting silicon (h) molded silicon with labels (i) final molded silicon	188

List of Tables

2.1	Existing Gripper Technical Specifications	22
4.1	Joint Speeds During Real-World, Calibration Grasps	81
4.2	Joint Speeds During Simulated Calibration Grasps	81
4.3	Difference in joint speeds during real-world and simulated calibration grasps	81
4.4	Chi-squared analysis of results, part 1 of 2. Where ' <i>delay</i> ' is the grasp time minus the optimum time, ' <i>n</i> ' is the number of tests conducted, χ^2 is the chi-squared value and P_{val} is the probability of obtaining the results presented if the strategy used has no effect on grasping performance. A single asterisk indicates a significant difference in performance at a significance level (α) < 0.05, a double asterisk indicates a significant difference in performance at $\alpha < 0.01$	88
4.5	Chi-squared analysis of results, part 2 of 2	89
4.6	An analysis of the effect of a reactive grasping strategy on the required spatial precision to achieve a successful grasp. The required spatial precision is defined as the largest spatial offset which results in a grasping success rate greater than 10%	93
5.1	Results of experiments to validate the repeatably of controlling and measuring the object speed. 600 tests were conducted at 3 different speeds . .	114
5.2	Summary of grasp performance across all test conditions for each control strategy, where 'Delay' is the delay after passing the light gate before the grasp is initiated for the 'Predictive' condition,	124

5.3	95% confidence intervals, comparing the performance of the reactive grasping strategy with that of the predictive strategy at 0ms, 5ms, and 10ms temporal offsets.	124
5.4	Summary of results from grasping experiments comparing a reactive strategy with a predictive strategy. Where 'Spatial Offset' is the distance from the centre of the gripper to the point on the finger where the object makes first contact, and 'Grasp Timing' is the delay after passing the light gate before the grasp is initiated for the 'Predictive' condition. An asterisk indicates a significant difference in performance at a significance level (α) < 0.05 . . .	125
6.1	Comparison of the grasp performance of the Neural Network and Predictive strategies across all test conditions. Where 'Grasp Timing' is the delay after passing the light gate before the grasp is initiated for the 'Predictive' strategy	147
6.2	Comparison of the grasp performance of the Neural Network and Reactive strategies across all test conditions	147
6.3	Summary of results from grasping experiments where the gripper is controlled by a neural network. Where 'Spatial Offset' is the distance from the centre of the gripper to the point on the finger where the object makes first contact, and 'Grasp Timing' is the delay after passing the light gate before the grasp is initiated for the 'Predictive' condition. An asterisk indicates a significant difference in performance at a significance level (α) < 0.05 . . .	148
6.4	95% confidence intervals, comparing the performance of the neural network grasping strategy with that of the predictive strategy at 0ms, 5ms, and 10ms temporal offsets, and the heuristic reactive strategy.	149

Nomenclature

k_p	Dynamic Stiffness
k_d	Dynamic Damping
m/s	Meters per Second
deg/sec	Degrees per Second
σ	Standard Deviation
Δ	Difference
ODE	Open Dynamics Engine
PI	Proportional Integral
DoF	Degrees Of Freedom
SDT	Standard Density Tactile
HDT	High Density Tactile
2D	Two-Dimensional
3D	Three-Dimensional
PbD	Programming by Demonstration
kNN	K-Nearest Neighbours
SMV	Support Vector Machines
ANN	Artificial Neural Networks
DNN	Deep Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
FFNN	Feed Forward Neural Network
SNN	Spiking Neural Network
RL	Reinforcement Learning
SL	Supervised Learning
FDM	Fused Deposition Modelling
PLA	Polylactic Acid
DC	Direct Current
IOT	Internet of Things

Glossary

Actuator

A device that causes a machine or other device to move.

Degrees of Freedom (DoF)

The number of specific, defined modes in which a mechanical device or system can move.

Therefore also the number of independent parameters which describe its configuration.

Manipulator

A robotic system, similar to a human arm, which consists of multiple links and joints and is used to position an end-effector such that it can achieve some task.

End-effector

The mechanism at the end of a robot manipulator tasked with interacting with the environment in some way

Gripper

A subset of end-effector, a gripper is a robotic system which is capable of grasping objects

Prehensor

Similar to gripper, a prehensor is part of a robotic system which grasps

Occlusion

A condition whereby an object of interest is not in line of sight of the robot's image sensors.

Morphology

Refers to the form or structure of the robot

Embodiment

The realisation of a robotic system

Pose

A vector describing the position and orientation of one coordinate frame relative to another

Exocentric sensor

A robotic sensor which is placed in the robot's environment

Egocentric sensor

A robotic sensor which is placed on-board the robot.

Proprioceptive sensing

A type of robotic sensing which specifically refers to measurements taken of the robot's environment

Exteroceptive sensing

A type of robotic sensing which specifically refers to measurements taken of the robot's internal state

Underactuation

Type of actuation system where the number of degrees of freedom of the robotic system is greater than the number of actuators

Phalanx

Part of a robotic finger, refers to a single link in the kinematic chain that makes up a robotic finger

Autonomous

Exclusively controlled by the robot, without human input

Kinematics

Describes the way in which a robot system moves, without consideration of its mass or the forces acting upon it

Anthropomorphic

A system which takes a similar form to a human. In the context of grasping often used to refer to robotic grippers which have similar kinematics to a human hand

1 | Introduction

Advances in robotic technology have enabled robots to be increasingly effective in a growing range of applications, increasing the ubiquity of the technology. Having already been revolutionary for automation in industrial applications, the capability of robotics has grown such that their usefulness has transcended applications in traditional manufacturing plants and factory assembly lines. Furthermore, there is demand to continue to improve robotic technologies and to apply robotics in new environments.

A major hurdle in the current state of the art of robotics is performing effectively in unstructured environments. Even relatively primitive robotic platforms perform well in highly-structured environments, where the structured nature of the environment can be leveraged to reduce the complexity of the task. Robots which work in less structured environments, like homes, shops, and even in flexible manufacturing settings, require much more complex control strategies. In order for a robot to be effective in a wider range of settings, new methods must be developed which increase their flexibility and adaptability and enable them to operate under dynamic, less structured conditions.

One area where this transition to more unstructured and dynamic environments is particularly challenging is grasping. Grasping is an essential function of robot platforms in a wide range of applications, extending the robot's ability beyond sensing and navigating its environment to include the ability to interact with its environment. In a controlled environment grasping a known object, in a known location and favorable orientation, is relatively simple. In contrast, a robot operating in unstructured or semi-structured environments needs to be able to grasp a wide range of previously unseen objects in a

wide range of possible positions and orientations. Finally, it is naive to assume that target objects will be static and remain static throughout the interaction. This thesis outlines research which specifically examines the autonomous robotic grasping of dynamic objects.

1.1 Motivation and Challenges

There has been a lot of research interest in improving grasp robustness in unstructured environments [1]. Prior research has focused on tasks involving changing lighting [2], objects which the robot is unfamiliar with [3], and in settings with occlusions [4, 5], relatively little research has addressed the fundamental aspects of grasping moving objects. Despite the lack of attention which this task has received from the research community, the ability of a robot to interact with objects in motion greatly increases the applicability of robotic technology. This is true both in industrial settings, such as on assembly lines where the emergence of flexible manufacturing has driven major improvements in productivity, as well as in nontraditional robotic settings, such as homes, hospitals, and logistics where new applications for mobile manipulators are emerging.

Current state-of-the-art grasping approaches for grasping moving objects have broadly followed the same approach, whereby the robot tracks the objects trajectory, predicts an appropriate interception position and time, moves its end-effector to that interception point, and attempts to grasp the object at that interception time using a predefined grasping motion. This predictive approach has shown some noteworthy results [6, 7], however, there are several issues that limit the applicability of these methods to real-world use-cases. One major contributor to the limitations of this approach is the heavy reliance on far-field, vision sensing to provide a gripper-object intercept position and time. Prior examples have been achieved almost exclusively in a laboratory setting where lighting, occlusion, and other environmental conditions were tightly controlled. Furthermore, gaining the required accuracy at the point of interception often necessitates the use of distributed motion tracking hardware which requires an existing infrastructure of sensors,

is costly and rarely practical in everyday settings. Finally, the high bandwidth, and the complex modeling and control required to accurately track, model, predict and intercept the object using vision sensing requires high computational effort.

These issues, among others explored in the following chapters, limit the usefulness of current approaches. Platforms with limited computational power, relying solely on on-board sensing, and operating in unstructured, unknown environments are at a significant disadvantage when relying solely on vision sensing and predictive strategies when grasping moving objects. Moreover, the potential contribution of other types of sensing and adaptive grasping motions remains under explored.

1.2 Research Objectives

The aim of this project is to explore how adaptive grasping motions, informed by real-time tactile feedback, can improve grasp robustness when grasping moving objects. Achieving this goal necessitates the development of tools and procedures, which quantify the performance of multiple different control strategies for grasping a moving object under a range of grasping conditions. Insights gained by exploring the performance of traditional strategies under non-ideal conditions, can be used to develop strategies which leverage real-time, tactile feedback to create reactive control strategies which adapt their grasping motion during an attempted grasp.

It is hypothesised that an adaptive grasping motion, based on real-time tactile sensor feedback, could mitigate the effects of errors in the interception of a ball by a robotic gripper, resulting in a more robust control strategy for grasping moving objects.

An experimental methodology is developed, which enables a systematic examination of the performance of different grasping strategies under a range of interception conditions. Simplifying the grasping problem allows the effect of specific parameters of the interception conditions to be examined. This includes reducing the problem from a three-dimensional to a two-dimensional problem, and removing the effect of object

orientation by using a spherical object. Using this methodology, the performance of a proposed strategy can be quantified and compared to the benchmark performance of the traditional strategy, thereby highlighting the associated strengths and weaknesses of such an approach. Finally, the suitability of machine learning techniques was explored and a machine learning agent developed, trained and tested. The performance of this machine learning control strategy was compared to the performance of the previously proposed reactive control strategy and the traditional control strategy.

In summary, the objectives of this thesis are to:

- Develop an experimental methodology to systematically quantify the performance of grasping strategies under a range of grasping conditions.
- Develop a simulated robotic gripper and testing environment capable of implementing the aforementioned methodology.
- Design and manufacture a robotic gripper and testing apparatus to replicate the simulated testing in the real-world utilising the same experimental methodology.
- Explore the performance of traditional control strategies and examine their robustness to different grasping conditions.
- Propose and explore strategies which leverage tactile sensor feedback to create adaptive grasping motions which react to grasping conditions in real-time.
- Analyse the performance of reactive control strategies and compare their performance to that of traditional control strategies.
- Design and develop a control strategy based on a machine learning agent and train that agent to create an optimised adaptive grasping motion based on real-time tactile sensor feedback.
- Quantify the performance of the machine learning agent compared to other control strategies.

1.3 Research scope

The task of autonomously grasping a moving object with a robotic platform, is a complex and multifaceted problem. The traditional approach to tackling this challenge can be represented as five stages, as illustrated in Figure 1.1. Though improvements in grasping performance may be achieved through innovation in any of these stages, this research has focused on the control strategies used in the final stage, ‘Intercepting the object’. This aspect of the problem has been under-explored and is subsequently an area with high potential for achieving performance improvements.

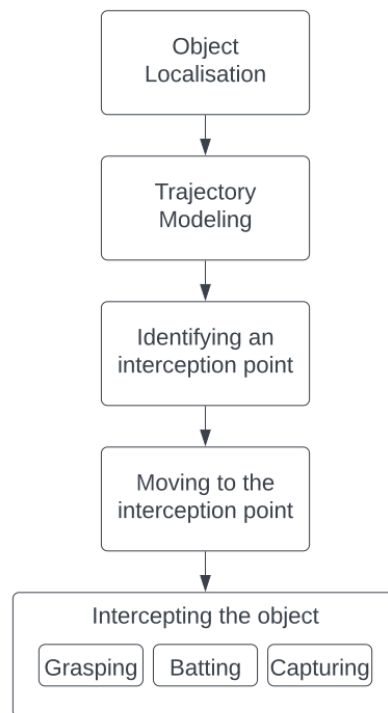


Figure 1.1: Flow chart outlining the stages taken by a robot autonomously interacting with a dynamic object

1.4 Contributions

This thesis contributes to the field in several discrete ways, including:

1. The design, realisation and validation of an experimental procedure which systematically quantifies the performance a control strategy for autonomously

grasping moving objects under a range of grasping conditions.

2. The development of a simulated testing environment in the robotics simulator Gazebo, which has been shown to provide accurate simulation of a gripper grasping a moving object. This simulated environment is capable of quantifying the performance of a grasping strategy under a range of grasping conditions and training a neural network which is suitable for deployment on a real-world gripper.
3. A real-world robotic gripper and testing apparatus which can systematically quantify the performance of different grasping strategies under a range of grasping conditions.
4. The characterisation of a traditional control strategy for grasping moving objects, specifically examining the effect of object speed, position of the initial contact between gripper and object, and grasp timing.
5. The proposal of a heuristic based reactive control strategy which leverages real-time tactile feedback to create an adaptive grasping motion, and has been shown to mitigate the effects of errors likely to occur in real-world conditions. Furthermore, it is effective when executed on a computationally constrained device and is not subject to sources of error common to vision sensing, i.e. occlusion, poor lighting conditions, etc.
6. Evidence of the contribution tactile sensing can make to the task of grasping moving objects, in particular to providing low latency, low bandwidth, local sensor data which can be used to adapt the grasping motion leading to more robust grasping.
7. Demonstration of the suitability of machine learning techniques to this task. This is achieved by demonstrating a neural network capable of achieving adaptive grasping behaviour, and high grasping performance on a real-world gripper.

1.5 Overview

The remainder of this thesis is structured as follows. Chapter 2 details relevant, existing research, this includes but is not limited to robotic grippers, robotic sensing, grasping moving objects, robotic simulation, and machine learning. Chapter 3 develops an experimental methodology which can systematically examine and quantify the performance of gripper grasping a moving object under a range of grasping conditions. Chapter 4 describes a simulated gripper and environment, as well as a series of tests conducted in simulation which quantify and compare the performance of a traditional and a reactive control strategy. Chapter 5 outlines a series of experiments conducted in the real-world. This includes the design and fabrication of the gripper used for testing throughout this research, as well testing which compares the performance of a heuristic-based reactive grasping strategy and a predictive grasping strategy deployed in the real-world. Chapter 6 examines the suitability of a neural network, trained using machine learning techniques, to the task of generating an adaptive grasping motion from real-time, tactile sensor feedback. Finally, chapter 7 summarises and draws conclusions from the work presented as well as identifying areas for future work.

2 | Literature Review

This chapter examines and discusses existing research and techniques regarding autonomous interactions with dynamic objects. It is divided into six sections. The first section examines the current approach taken to interacting with dynamic objects, and contextualises the research in this thesis within that process. Next, existing robotic grippers, discussing characteristics such as their morphology, design and actuation systems. Robotic sensing is addressed next, specifically focusing on tactile and joint position sensing. The gripper and sensing sections provide the background information necessary to understand the robotic systems which are central to this research.

Furthermore, it will help the reader to understand many of the design decisions, both gripper design and experimental design, taken during the course of this research. Having established the typical approach to interacting with moving objects and the state of the art in existing robotic gripper and sensing systems, the next section will present and categorise a comprehensive list of examples of robots which have been tasked with autonomously interacting with dynamic objects. The penultimate subject addressed in this literature review is machine learning techniques, specifically those which have been shown to be effective in robotic grasping applications. Finally the use of simulation in robotics research is discussed with reference to examples from the literature.

2.1 Interacting with a dynamic object

This section will explore the manner in which robots typically interact with moving objects. The approach taken can be divided into five stages, shown in Figure 1.1. The

remainder of this section will examine these five stages more closely.

2.1.1 Vision sensing detects and localises the object

The requirement to track the far-field trajectory of the moving object is common to all examples of interacting with moving objects. This is achieved using a range of image systems including combinations of monocular cameras producing stereo images [8, 9, 10, 11, 12, 13, 14, 15], structured light cameras [16, 17], and specialized motion capture hardware [7, 18, 19, 20, 21, 22]. These systems are discussed in Section 2.3.2

A reliance on vision sensing is evident from the prior art, however this technique has many practical limitations and in particular is difficult to implement in unstructured environments, where issues such as occlusion and poor lighting conditions come to the fore. A reliance on exocentric sensing, i.e. sensors placed in the environment, is also evident. A significant drawback of this is the need for an existing infrastructure of sensors, which is not feasible in many environments. This indicates that the performance of current state of the art methods would degrade significantly when applied to exclusively egocentric sensing, i.e. on-board sensors, operating in unstructured environments [7].

2.1.2 Modeling and estimating object trajectory

Intercepting a moving object fundamentally requires some prediction of the future state of the object. In the case of a ball which is thrown to a robot by a human, a common example in prior research [6, 14, 17], the future path of the ball must be extrapolated from sensor data in real time so the robot can intercept it, see Figure 2.1. The simplest way this has been done in the past is to fit a parabola to observed ball positions and assume the object will follow a parabolic path [12, 23, 24]. Depending upon the properties of the thrown object this prediction might prove sufficient, though more accurate models also consider the effect of air resistance [6, 7, 8, 14, 18, 25] and the Magnus effect [26]. A theoretical analysis of the effects of drag on spherical objects [25],

showed that two models (one assuming no drag and one accounting for air resistance) which model a soft ball thrown through the air (initial velocity of $7m/s$ at a 45° angle) result in end positions which are 1 meter apart. In practical applications it is not always possible to know parameters like the dimensions and the mass of an object, which is a significant source of error when estimating the appropriate point of interception.

Other than attempting to directly model the physics of the object trajectory, there are several examples which attempt to learn the behaviour of the object, these approaches include genetic programming [10], neural networks [27], and k-nearest neighbours [9]. Objects which bounce or otherwise interact with the environment before the grasp add additional complexity to modeling the trajectory. These are discussed in more detail in other publications [28].

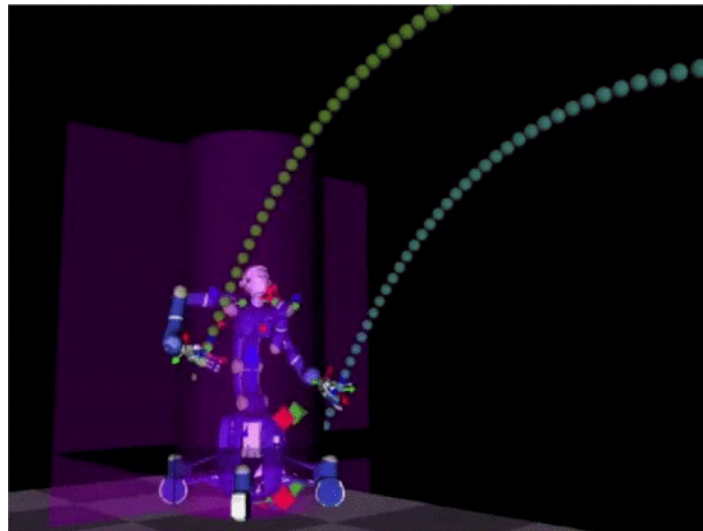


Figure 2.1: Illustration of Rollin Justin predicting the future trajectory of two balls simultaneously [6]

2.1.3 Identifying an appropriate interception point

In theory, a robot can choose to attempt to intercept the object at any point along the object's trajectory which intersects the reachable volume of the robot. Early research adopted a very simple approaches and choose the point along the objects trajectory which was closest to the robot base [29] or closest to the initial position of the hand [24]. Other examples added additional constraints to the choice of an interception point, including

requiring a manipulability parameter to be above a certain threshold [19], or specifying an allowable interception region which is not too close and not too far away from the robot [25]. Some examples have limited the interception point selection to a plane by first defining a desired interception height [8, 15, 23] or an interception window in front of the robot [12]. From these the interception point is where the object's trajectory is at the target height or in the interception window. More recent work has combined the tasks of selecting an appropriate interception, calculating a catch configuration and generating a path of the robotic manipulator into a single, nonlinear optimization problem with nonlinear constraints. This accounts for joint position and acceleration limits, as well as geometrical limits to avoid self collision and work cell limits [6, 14].

2.1.4 Moving the end-effector to the interception point

The end-effector must reach the estimated interception position before the object passes this point and the opportunity to intercept is missed. A common implementation uses the estimated interception position and time to calculate the necessary joint movement using inverse kinematics [7]. Machine learning methods have also been applied to this. Inverse dynamics learning has been used to achieve the high speed manipulator motion necessary for ball catching [18]. This process is also prone to inducing errors in the interception. A robotic manipulator and gripper are typically long, open kinematic chains. Small errors in the joint angles can propagate through the chain leading to a larger error in the position of the end effector.

2.1.5 Intercepting the object

The final stage of interacting with a moving object is intercepting the object. Intercepting a moving object is any interaction which involves physical contact between the object and robot end effector. This can take several forms, including batting, hitting, capturing and grasping. A hitting and batting style interception is the simplest type of interception, a specialised robot end effector contacts the object in order to effect its trajectory in some way, then allowing the motion of the object to continue. Capturing a dynamic object is

more complex in that it attempts to bring the object under control, the end result should be zero relative motion between the robot end effector and the object. Examples of capturing stops short of grasping the object. This research is specifically concerned with grasping the object during the interception. Grasping in this context refers to actuating a robotic gripper during the interception of a moving object to secure the object.

An important consideration is when to initiate the grasping motion. The predominate way used to determining grasp timing is to use the relative positions of the object and end-effector. The grasp is either triggered when the distance between gripper and object drops below a tuned threshold [23, 28, 29, 30, 31], or using object tracking and prediction methods to estimate a 'catch frame' [14, 32], i.e. a point in time at which the object will be in a graspable position. Existing research has explored how the distance between gripper and object can enable a coupled dynamic systems approach where the joint positions of the gripper are coupled to the 'distance to gripper' metric [33].

Examples from the literature use predictive control strategies to achieve a successful grasp. A predictive control strategy relies solely on the estimated interception position and time, moving the gripper into position and executing a predefined gripper closing motion at the estimated time. From the gripper controller's perspective this strategy is entirely predictive as estimates of object trajectory and joint position from the previous stages are utilised to formulate the gripper action rather than real time sensing of the object position during the grasp. Though predictive strategies have shown some noteworthy results [6, 7], several issues limit the applicability of these methods to real-world, unstructured settings. Firstly, results have been achieved in a laboratory setting where lighting and other environmental conditions were tightly controlled to optimise the performance of vision sensors. Secondly, performance has relied almost exclusively on the use of exocentric motion capture sensors, and are subject to all of the associated disadvantages previously discussed. Besides potential errors in sensing, modeling the future trajectory of the object and choosing an appropriate interception point is also challenging. Unknown object properties, inaccurate models and the demands of rapidly computing complex optimisation problems can all contribute to errors in the estimated interception position and time.

Even assuming perfect estimation of an interception point, rapidly moving the end effector to that position while controlling its precise location is difficult. The aggregation of these errors in sensing, modeling and actuation leads to misalignment between the actual and optimum interception. A predictive control strategy has no mechanism to attempt to mitigate the effects of these errors, and there is a lack of research which addresses grasping motions which attempt to adapt to these errors.

2.2 Existing robotic grippers

The review of existing robotic grippers outlined in this chapter provides foundational knowledge to the project. An examination of these grippers informed the design and manufacture of the gripper developed for this research, described further in later chapters.

There is a large and diverse range of robotic grippers, currently in use both in research and industry. These grippers vary hugely, from simplistic to complex, and from general purpose to specialised embodiments for specific, niche applications. It would be difficult to complete a comprehensive review of all available grippers, instead this list has been tailored to a representative list of the most relevant examples. More extensive reviews of existing robotic grippers have been conducted by other researchers and typically focus on a specific gripper type, application or technology. For example, literature reviews have been conducted on dexterous hands [34], on parallel, two-finger, industrial grippers [35, 36], soft manipulators and grippers [37], anthropomorphic hands [38], underactuation [39], and grasping in the context of agricultural applications [40].

This section will discuss grippers under several different categorisations. These groupings are commonly used to describe grippers [41], however due to the complex nature of these robotic systems it is possible for grippers to fall into more than one category.

Low degree of freedom prehensors

These are the simplest type of gripper available and include examples like the 2F-85 model by RobotIQ [42], the RG2 gripper by Omron [43], and the end effector on the KUKA youBot [44]. These are shown in Figure 2.2. Their low complexity and large application range make this type of gripper very common in industrial settings where the environment and task are highly structured. A common embodiment of this type of gripper is the pincer gripper, which uses two opposing fingers to apply a clamping force onto the object. It is common to further reduce the complexity by using coupled joints or by using one static finger and one actuated finger. They require low control complexity but are inherently limited and can encounter problems if the object is too large, heavy or irregularly shaped.



(a)



(b)



(c)

Figure 2.2: Sample of existing low prehensor grippers, (a) Robotiq 2F-85 [42] (b) OnRobot RG2 [43] (c) Kuka youBot platform [44]

3-finger grippers

As the name suggests, these robotic grippers have three fingers. Popular robotic grippers which fall into this category include the Schunk SDH gripper [45], the Barrett Hand [46], and the RobotIQ 3-Finger Adaptive Robotic Gripper [47]. These are shown in Figure 2.3. Due to the larger number of fingers, and therefore DoF, the complexity, both mechanical and control, is higher for these grippers compared to their lower DoF prehensor counterparts. However, the number of grasp configurations which are possible also increases. These grippers have more potential as general purpose robotic grippers which could operate effectively in an unstructured environment.

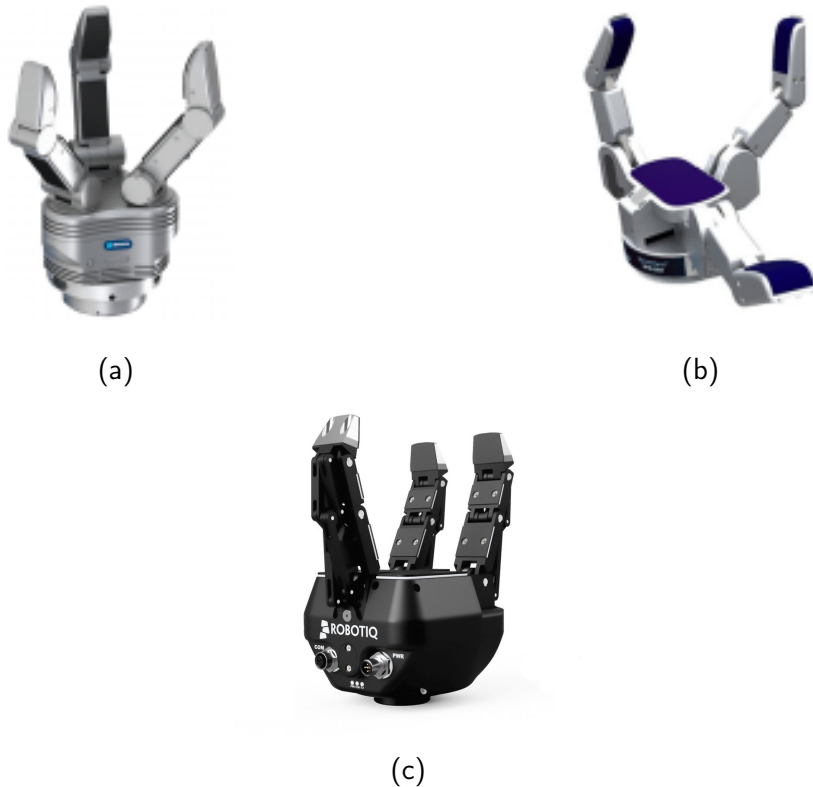


Figure 2.3: Examples of existing 3 fingered robotic grippers (a) Schunk SDH [45] (b) Barrett Hand [48] (c) Robotiq 3-finger adaptive gripper [47]

Anthropomorphic robotic hands

Anthropomorphic robotic hands have the essential characteristic that they are similar to the human hand and include examples such as the DLR Hand II [49], the Allegro hand [50, 51], the Shadow Hand [52, 53], and the Schunk SVH platform [54, 55]. As well as

the grippers implemented on the Twendy One [56], iCub [57], and James [58] robotic platforms.

Gripper morphologies which are similar to that of a human hand tend to operate well in unstructured environments [38, 54]. This can be attributed to the fact that the human hand is an excellent general purpose tool, but also because many of the unstructured environments which we wish robots to operate in tend to have been designed for humans. They are therefore generally well suited to robotic grippers which are of a similar size, shape, and dexterity to a human hand. There are other motivations to use an anthropomorphic gripper also. Their similarity aesthetically to a human hand makes them very desirable for prosthetics and they are advantageous for applications where a gripper is tele-operated by a human since joint angles can be mapped directly from one to the other [38].

A disadvantage of the anthropomorphic gripper is their complexity, exceeding that of the previously examined categories. This extends to their mechanical design, which must accommodate typically tens of degrees of freedom (DoF) in a compact assembly, as well as the sensing and control requirements. Furthermore, the additional complexity impacts the cost and reliability of the gripper.



(a)



(b)



(c)



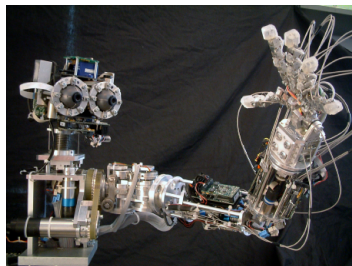
(d)



(e)



(f)



(g)

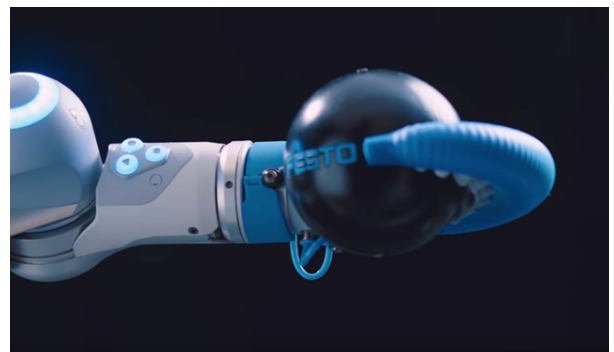
Figure 2.4: Sample of existing anthropomorphic robotic grippers. (a) DLR-Hand-II [59] (b) Allegro Hand [60] (c) Shadow Hand [61] (d) Schunk SVH [62] (e) TWENDY ONE [63] (f) iCub [64] (g) James [58]

Soft robotic hands

Soft robotic grippers distinguish themselves through the use of compliant materials in the place of rigid links, as used in traditional robotic grippers. Examples include particle jamming grippers [65], the OctoGripper [66], and the RBO Hand 2 [67]. These examples can be seen in Figure 2.5. The introduction of soft, compliant components into the design required a completely different set of control principles. This can lead to simpler control, and higher adaptability though this often comes at the cost of dexterity. Common applications of this type of gripper include handling fragile objects, and human robot interaction, where the soft materials help ensure the safety of the human. The high levels of inherent compliance as well and the actuation mechanisms which are typical slow relative to other types of gripper tend to make soft robotic grippers unsuitable for grasping moving objects.



(a)



(b)



(c)

Figure 2.5: Examples of existing soft robotic grippers. (a) Particle jamming gripper [65] (b) OctoGripper [66] (c) RBO Hand 2 [67]

Vacuum Gripper

Vacuum Grippers are common and in certain applications can be highly effective, while limited, in the range of objects they can effectively interact with [68, 69]. Team Delft, winners of the 2016 Amazon Picking Challenge, used a hybrid pincer and vacuum gripper [70]. An example of a commercially available vacuum gripper is the VGC10 platform from OnRobot [71], shown in Figure 2.6



Figure 2.6: Example of an existing commercially available vacuum gripper

2.2.1 Underactuated robotic grippers

Underactuation is a popular technique used to actuate grippers. It is typically used for a few reasons, it reduces the number of actuators necessary to control the gripper, it simplifies the control problem and it can take advantage of the systems natural dynamic behaviour.

Underactuation refers to when the gripper has more joints or DoF than actuators. A more formal definition can be constructed starting from Newton's laws of motion. Newton's second law $F = ma$ implies that the acceleration of an object can be described by a second order function i.e. if the position and velocity of an object is known as well as the forces acting on it, its acceleration can be calculated. Equation 2.1 shows the general form for a second order dynamic system [72].

$$\ddot{q} = f(q, \dot{q}, u, t) \quad (2.1)$$

where:

u is the control vector

q is a vector describing the robot's position

t is time

$\dot{q}$ is a vector describing the robot's velocity

For most robots, this becomes

$$\ddot{q} = f_1(q, \dot{q}, t) + f_2(q, \dot{q}, t)u \quad (2.2)$$

From this definition, the system is said to be underactuated if

$$\text{rank}[f_2(q, \dot{q}, t)] < \dim[q] \quad (2.3)$$

The result of this is that a single actuator is used to actuate more than one joint, thereby coupling their movement. The popularity of this technique can be seen in Table 2.1 where the technical specifications of some of the aforementioned grippers is tabulated.

Although this gives a useful summary of popular robotic grippers, direct comparison in this way is difficult and there are several caveats to be aware of when using this table.

Some of the differences between these grippers include:

- The mechanism by which the joints are coupled, ranging from simple gears to novel, patented mechanisms
- The choice and placement of actuators. Some grippers emphasise the importance of integrating the motors into the gripper while others place the motors in the wrist or elsewhere on the robot
- The number of degrees of freedom (DoF). Several of the grippers are part of a larger manipulator or robotic system and therefore the wrist joints are a fundamental part of the gripper system, while others isolate the gripper and require integration into a separate robotic system, resulting in a lower stated number of DoF but in reality have very similar gripper dexterity.

Name	Fingers	Joints	Actuators	Underact.	Example of Use
Sky	5	5	Unknown	Yes	[17]
GIFU Hand III	5	20	16	Yes	[73, 74]
DLR-Hand-II	4	13	13	No	[6, 14, 49]
TWENDY ONE	4	16	13	Yes	[56, 75, 76]
Allegro Hand	4	16	16	No	[7, 22, 51, 77, 78]
iCub	5	19	9	Yes	[7, 79]
Schunk SDH	3	7	7	No	[16]
Barrett	3	7	4	Yes	[80, 81]
Shadow	5	24	20	Yes	[53, 82, 83, 84, 85, 86, 87]
Schunk SVH	5	20	9	Yes	[55]
Obrero	3	11	8	Yes	[88, 89, 90]

Table 2.1: Existing Gripper Technical Specifications

2.3 Sensing for robotic grippers

Many examples of robotic platforms autonomously interacting with dynamics objects seen in the literature exclusively rely on vision sensing and so the potential contribution of other types of sensing modalities to this task remains underexplored. This section examines several different types of common sensing on robotic grippers and aims to give the reader the necessary background to understand the context of this research.

This section will focus specifically on aspects of robotic sensing which are most relevant to this research and is broken into four sections. First, a review of the joint angle sensing is presented. Next vision sensing is examined, this includes a review of the different mechanisms used in vision sensing and examples of when vision sensing has been previously used in robotic grasping. Next, tactile sensing is addressed, including common techniques used to in tactile sensors and examples of tactile sensing in research are given. The final part of this section discusses visuotactile systems, discussing instances in prior research where the combination of vision and tactile sensing have been used effectively.

2.3.1 Joint angle sensing

Joint sensing is an example of proprioceptive sensing, enabling the robot to monitor the position of each of its joints and thereby extrapolate its own pose in space. It is an

essential component in the accurate control of any gripper. There are two mechanisms which are commonly used to measure joint angles, these are described below.

Potentiometers

Potentiometers are a commonly used tool for collecting joint angle data [49, 56, 88]. Potentiometers consist of a resistive element with a wiper that moves along it and is connected to the output of the sensor. The resistance between one end of the resistive element and the wiper therefore changes as the wiper moves. This can readily be converted to a voltage for data acquisition. For rotary potentiometers the output resistance is determined by the relative orientation of an internal shaft to the rest of the sensor. By coupling this internal shaft to a joint, the joint angle can be measured. A drawback of this type of joint sensor is that the potentiometer will have some maximum and minimum resistance which it can achieve and is therefore unsuitable for measuring continuous rotation. In reality potentiometers either have a finite angle of rotation or have a discontinuity in their resistance at some angle. This is generally not a problem for robotic grippers or manipulator joints which also tend to have a finite joint angle range and do not require continuous rotation.

Encoders

Encoders measure relative rotation between two parts of a robotic system. Absolute encoders measure the angle in absolute terms while incremental encoders monitor the relative movement. There are several different mechanisms which are used to achieve this including optical, mechanical and magnetic. Absolute encoders are more commonly used on grippers [45, 50] since it is the absolute position which is necessary to calculate the position of the gripper, however there are examples of the use of incremental encoders in grippers [57]. Magnetic encoders are particularly common because of the contactless nature of their implementation [46, 52, 57, 73]. The relative orientation of two parts of the robotic system are monitored by attaching a magnet to one and a Hall effect sensor to the other. The measured magnetic field is dependant on their relative orientations and the

position of the joint can be inferred.

2.3.2 Visual sensing in robotic grasping

Vision sensing is the most popular sensing modality used in grasping. It provides high fidelity data about the object and the environment remotely and without physical interaction. The popularity of vision sensing is clearly visible when examining examples of robots autonomously grasping moving objects. All examples examined later in this chapter (see Section 2.4) use vision sensing in some capacity. Though the contribution of vision sensing to the task of grasping moving objects is invaluable, there are some fundamental issues with relying solely on vision sensing, particularly when operating in unstructured environments, which this research attempts to address.

Image sensing can be broken into two different categories; two-dimensional (2D) and three-dimensional (3D). Both collect images of the environment while 3D sensing adds depth data. Furthermore, both types of sensing have been used in prior examples of robots interacting with dynamic objects.

The simplest implementations involve using combinations of multiple 2D, monocular cameras to produce stereo images. This technique, known as "stereo pairs", operates using the same mechanism as the human eyes. A pair of traditional 2D image sensors are placed a known distance apart. Using the differences between the two images collected and basic geometric calculations, depth information can be inferred. Stereo pairs have been used to enable the Rollin Justin platform to grasp two balls which are simultaneously tossed to it [6], to guide a 6 DoF arm to catch balls thrown from 4m away in a cup [12], to track a workpiece on a conveyor belt [13], and examine the kinematically optimal way to catch a ball using a DLR-LWR-III robotic manipulator and DLR-Hand-II robotic gripper [14].

Researchers have also tracked the motion of moving objects using one or more 3D vision sensors, such as structured light cameras [16, 17]. Structured light cameras project IR light patterns of a known structure onto the environment and extract depth information

from how the patterns interfere with the scene. This mechanism is used in commercial depth cameras like the Microsoft Kinect and Asus Xtion [41]. In a study by Cuevas-Velasquez et al., the researchers achieve dynamic object tracking under partial occlusions using four structured light cameras to track the far-field motion of the object of the object and a robot-mounted stereo camera to track the near-field motion [5].

The most sophisticated object tracking setups have used specialised motion capture hardware, designed specifically to track the motion of objects moving in 3D space. For example the Optitrack motion capture system from Natural Point [91]. The Optitrack system has been used enable the KUKA LBR IIWA, 7 degree of freedom robot arm and Allegro Gripper to catch fast moving objects by matching the velocity vector of the object to reduce the relative speed between gripper and object [22]. It has also been used to enable the KUKA LWR 4+ and Allegro Gripper to catch irregular objects which been tossed toward them [7]. Similar motion capture systems have been used in several existing examples of robots autonomously grasping moving objects [7, 20, 21, 22]. While these systems boast the greatest accuracy and reliability, they are expensive and require a complex environment setup.

There are several limitations to the current approach of applying image sensing to the task of grasping moving objects. These are:

- Large quantities of sensor data

One such challenge is the large quantities of data which it generates. This data puts enormous pressure on the system's bandwidth and the computational systems which must transport and process this data. Though this is considered a major strength of vision sensing in some applications, this is a huge problem in others.

- High latency

The impact the large quantities of sensor data are amplified when the task is time critical, as is the case while grasping moving objects. It can be difficult to read from the sensor, process the image data, and act quickly enough in these applications.

This is particularly true if the robotic platform is computationally constrained.

- Occlusion

Occlusion is another well documented challenge associated with vision sensing. This refers to situations in which vision sensing proves to be ineffective because there is an obstacle between the sensor and the target object.

- Sensitivity to lighting conditions

Image sensing is also sensitive to lighting conditions with many implementations struggling to remain consistent under different lighting conditions. Poor lighting conditions can significantly limit the usefulness of vision sensing, in particular in an unstructured environment [4, 92].

- Image sensor placement

The placement of image sensors in prior examples of grasping moving objects should not be overlooked. To the best knowledge of the author, only one study has accomplished this task using exclusively egocentric sensing, where the only sensors used to track the moving object were located on-board the robot [6]. With few exceptions, object tracking has been achieved primarily by using an arrangement of cameras in the area surrounding the robot. By using exocentric camera systems more accurate localisation, tracking, and modeling of the objects trajectory can be achieved. This setup and the associated high levels of accuracy can not be achieved in unstructured, real-world conditions.

2.3.3 Tactile sensing in robotic grasping

Tactile sensing refers to the robot's ability to detect contact between the robot and an object. Research has found a high reliance, by humans, on a "sense of touch" for many common tasks [93, 94]. Roboticians have long thought that robots could benefit from "sense of touch" in the same way as humans. Tactile sensing has the potential to provide a deeper understanding of the nature of the interaction between robot and objects would allow them to better understand and control their interactions in the real world.

Tactile sensing is used in a wide variety of tasks and is particularly effective in tackling

uncertainty arising from unstructured environments. Applied to robotic grasping it can help a system infer information about an object's properties like size, shape, weight, compliance, texture and temperature [78, 87, 95, 96, 97, 98, 99]. This is essential for detecting slip between gripper and object [83, 100, 101, 102, 103], and can also be used to inform and control robotic motion, referred to as tactile servoing [104, 105, 106]. It is therefore useful, not only in grasping tasks [107, 108, 109] but also in object exploration [106, 110, 111, 112], object classification [76, 77, 85, 87, 97, 113], object localisation [95, 114], assessing grasp stability [115, 116, 117] and in-hand manipulation [75, 86].

To date, examples of robots autonomously interacting with dynamic objects have relied heavily on vision sensing, requiring highly accurate object tracking, optimum conditions and often external sensors. Tactile sensing offers real-time, low bandwidth, localised information about the position of the ball which could potentially enable an adaptive grasping strategy. This potential has yet to be explored.

Despite widespread agreement in the robotics community that tactile sensing is essential for robots operating in unstructured environments [1, 95, 118], there is no one sensor or sensing mechanism which meets the needs of the wide range of robotic applications which use tactile sensing. This has led to an enormous range of existing sensors, each developed to meet a niche application while compromising elsewhere. Several detailed reviews have been produced which document the types and applications of tactile sensing for robotics [118, 119, 120, 121, 122]. What follows is a summary of the most pertinent tactile technologies to this work

Magnetic

Tactile sensing based on magnets utilises a magnetic field, measured by a Hall effect sensor to determine information about contact between robot and object[123]. Typically in a Hall effect based tactile sensor the position of a magnet relative to a Hall effect sensor is in some way related to the current state of contact between the robot and an object. By inferring details about the nature of contact, i.e. location, force, etc. from the

strength of the magnetic field to achieve sensitive manipulation, whereby the gripper is able to find, reach and grasp unknown objects using only tactile sensing [90].

Furthermore, some Hall effect sensors can read the magnetic field strength in multiple axes [124]. This additional functionality enabled an Allegro robotic hand equipped with the uSkin tactile sensor to estimate the weight of objects by lifting them [78]. Using the tactile sensing to detect forces tangential to the surface as well as normal forces also enables the sensors to be used to detect and prevent slip. This can be demonstrated on a prosthetic hand using magnetic tactile sensing to grasp deformable objects [102].

The aforementioned uSkin is a particularly popular implementation of this type of tactile sensing and has been applied to several grippers, including the end-effector of a Sawyer robot [125], the iCub gripper [57], and the Allegro hand [50, 126] as shown in Figure 2.7.



Figure 2.7: uSkin curved product offered by xelarobotics, specifically designed to be used as the fingertip for the Allegro robot hand [127]. (a) shows a rendering of the sensor and (b) shows a photo of uSkin deployed on the gripper

Resistive tactile sensors

Resistive tactile sensing relies on materials whose resistive properties varies with applied force or pressure. Prior literature demonstrates several ways this can be used. A pressure conductive rubber was used to create a sensor which was able to achieve complex in-hand manipulation in the form of pen-spinning with a three fingered robotic hand [128].

Another example demonstrated object classification through stiffness estimation by using a piezoresistive fabric layer surrounding by two conductive layers [85].

Resistive tactile sensors can be implemented using strain gauges, force-sensing resistors, pressure conductive elastomer and pressure sensitive ink [122], and has been used to develop tactile sensors for a range of advanced robotic systems [85, 103, 128, 129, 130].

A subset of this type of tactile sensing which has gained traction in recent years is tunneling effect tactile sensors [120, 131]. These sensors rely on quantum tunneling composites, which are polymeric materials containing conductive nanoparticles. Compressive loads cause the volume of the insulating polymer to decrease and the localised density of nanoparticles to increase. Under the compressive load a tunneling conduction mechanism reduces the resistance of the material, enabling it to be used as a tactile device. This was one of the mechanisms used for tactile sensing on the Shadow robot hand [84].

Capacitive tactile sensors

Capacitive tactile sensing uses the physical phenomenon that the potential different between two conductors, separated by an dielectric, is dependant on the distance between them and the overlapping surface area. By using a soft, elastic material as an insulator between two conductors the capacitance will reflect the deformation of the elastic insulator and allow the robot to infer information about nature of contact with an object. This type of tactile sensor has proven very popular since they are easy to manufacture, can be created in very small sizes, and can be used to create high resolution sensing arrays. Their popularity is evident in the range of grippers they have been applied to including the fingertips of the robot iCub[79], the parallel jaw gripper on the PR2 robot[132], the Barrett Hand, and TWENDY-ONE [122]. There are also numerous commercially available capacitive sensors targeted at applications in robotics [133, 134]. Each of these examples use capacitive sensing to achieve high resolution tactile information with 12, 22, 96 and 241 sensing elements or sensitive zones for the iCub, PR2, Barrett and TWENDY-ONE grippers respectively.

Pressure tactile sensors

Pressure tactile sensors operate by using a barometric sensor to measure the pressure of a deformable material, usually rubber. The measured pressure of the material is proportional to the force exerted upon it through contact with an object. This is the principle used by the TakkTile tactile sensor [135], implemented on the ReFlex robot hand [136].

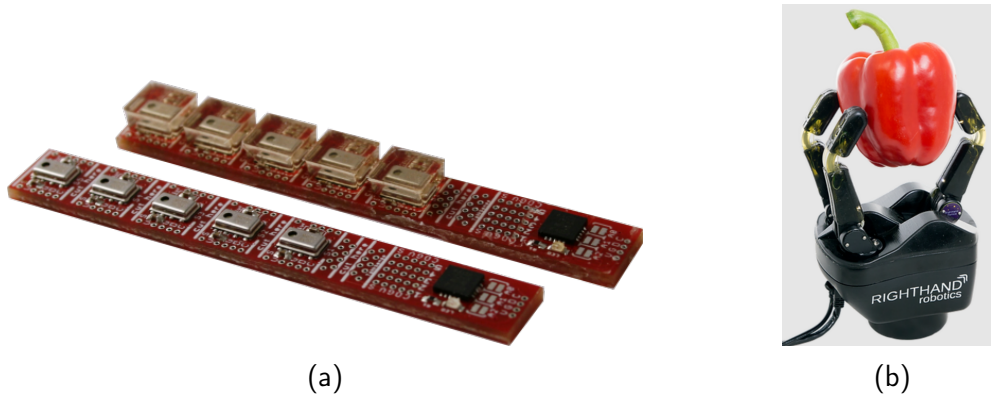


Figure 2.8: TakkTile pressure based tactile sensors by the Soft Robotics Toolkit, (a) shows the sensor itself [137], (b) the sensor implemented on the ReFlex robotic gripper [136]

Optical tactile sensors

Optical tactile sensors use the properties of reflected light to infer information about contact between sensor and object. There are several ways this can be achieved, using the wavelength reflected from a Bragg grating to determine the applied strain is one such means [138]. The Kinotex sensor offers an alternative implementation [139]. The intensity of light, interacting with an optical cavity, is monitored. The volume of the cavity is dependant upon contact forces which proportionally effect the sampled light intensity [140]. A third example is the OptoForce tactile sensor [141] which has been implemented on the RobotIQ three fingered gripper and used to detect slip in deformable objects [142]. This principle is also used to implement tactile sensing on the DEXMART hand [143]

Vision-based tactile sensors

These sensors aim to leverage the advancements and benefits of vision sensing and vision-algorithms to deliver high resolution, robust, inexpensive tactile sensing [110, 122]. Although techniques vary, the underlying principle is that an image sensor and vision algorithm will monitor the displacement of markers on the sensor surface caused by contact with an object [144, 145, 146]. From this, information about the contact can be inferred. There are several examples of tactile sensors which use this technique, these include the TacTip [106, 146], GelSight [145, 147], GelSlim [95] and FingerVision [144]. The TacTip sensor aimed to create a step-change in the resolution possible with a tactile sensor and was tested on a IRB 120, ABB Robotics manipulator [146]. Later the same sensor, manipulator combination was used for edge perception and contour following [106]. The GelSight has been implemented on the Baxter robot hand and used for precise object localisation within the grasp [148] as well as informing regrasping policies [147]. The FingerVision sensor was first demonstrated on a Baxter robot tasked with chopping fruit and vegetables [144]

Using proximity sensors as tactile sensor

Proximity sensing offers a mechanism which is subtly different to the other types outlined above, in that it is possible to sense the object without contact. There are several examples of proximity sensing being used as tactile sensing where zero distance to the object is assumed to be contact [21]. For the most part these types of tactile sensors have used time of flight, capacitive and optical mechanisms to approximate the distance to the object. These methods often suffer from a lack of accuracy, long measurement times and sensitivity to the surface properties of the target object. Significant progress has been made in this area using a principle based on analysing the phase shift of a reflected modulated light signals [149] which has recently been miniaturised and applied to a robotic gripper [150]

2.3.4 Visuotactile sensing in robotic grasping

On-board, real-time processing of multi-sensory data has been identified as a way to tackle many challenges associated with implementing robots in unstructured environments [1] and has been shown to be effective in a range of applications [108, 112, 147]. Prior research has shown progress through the combination of vision and tactile sensing [41, 81, 128]. It has been used to monitor the relationship between gripper and grasped object, e.g. during a drilling operation [151]. A combination of force, vision and tactile sensors has been used to tackle uncertainty in grasping static objects [92, 152] and was popular in several grasping challenges and competitions [41]. In fact there has been an explosion in recent years in research aiming to use machine learning techniques to process a combination of vision and tactile sensor data to complete a range of goals, including assessing grasp stability [153], inferring information about the objects surface properties [97], informing and improving a reattempted grasp of a static object [109], and to improve a robot's ability to identify and recognise unknown objects [77, 95]. These examples have all been concerned with static objects or objects where there is no relative movement between gripper and target object. A visuotactile sensing system has previously been used to aid in the grasping of moving objects [21], this is discussed more in the next section.

2.4 Examples of interactions with dynamic objects

Interacting with a dynamic object presents a significant challenge above that already presented by the interaction with static objects. The dynamic nature of the object increases the demands placed on sensing, computation and actuation systems. Despite these challenges, it has long been a topic of research in robotics with foundational work being conducted in the early 1990s [24, 154]. Since then, examples in research include but are not limited to robots playing games [155, 156, 157], catching tennis rackets [7], catching balls [6, 17, 22], batting [26, 158, 159] and juggling [23, 154, 160].

A novel analysis of the problem space is proposed in this section, where examples from the

literature of robots autonomously interacting with dynamic objects are plotted along two axes. The task complexity axis, ranging from simple to complex, rates the challenge level of completing the task outlined in the example with a robotic system. An important consideration for this axis is the structured or unstructured nature of the environment. Structured environments can leverage factors such as controlled lighting, lack of foreign objects moving through the area, sensor placement in the environment, etc. Structured interaction with the object is also considered in the task complexity. Examples where the degree of manipulation necessary is reduced have a lower task complexity. The second axis is an object axis, ranging from simple to complex. This describes the type of object which is grasped, including factors such as the shape of the object, whether an accurate model of the object is available, if the robot has previously interacted with the object, if the object has a regular weight distribution, etc. These examples are discussed below and are subsequently mapped onto these axis, is shown in Fig 2.9.

By applying the examples found in the literature to this framework several groupings within the problem space emerge. These groupings, which are discussed in more detail below, are; 1) where the object is on a conveyor belt, 2) hitting or batting an object, 3) capturing a ball, 4) grasping a ball and 5) grasping an irregular object.

2.4.1 Examples where the object is on a conveyor belt

This group occupies the area of the problem space where the environment is highly structured but object complexity can range from simple to complex. In examples of objects on a conveyor belt the environment is considered to be highly structured because the object's geometry, position, velocity and pose are known to a high degree of accuracy. Furthermore, the movement of the conveyor itself is designed to ensure that the robot can effectively interact with the moving object. Simply tracking the movement of the conveyor belt with a robotic manipulator can simplify the problem to a quasi-static problem, albeit a time-critical one. The object in these examples can range from the simplest of objects, i.e. a sphere, to objects with complex geometries which the robot has never encountered before.

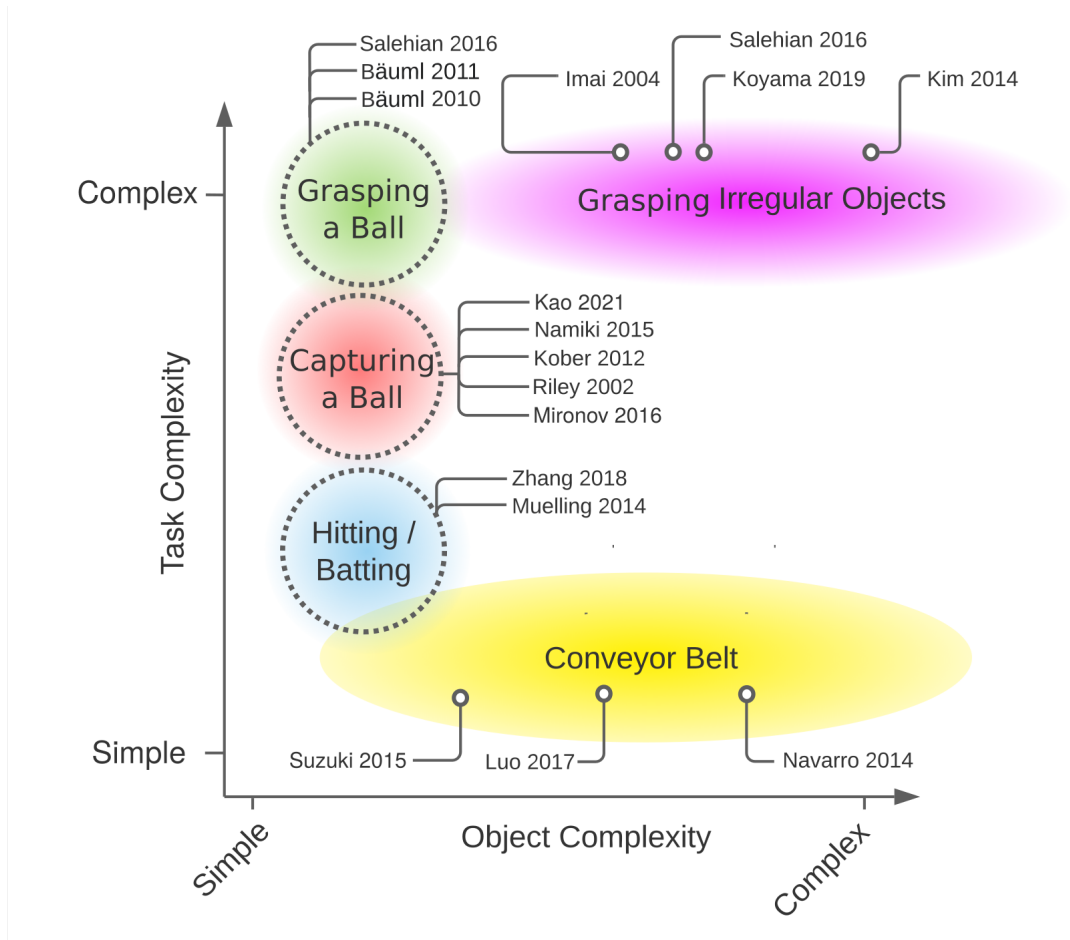


Figure 2.9: Sample of existing research of autonomous interaction between robot and dynamic object, mapped based on object complexity and the nature of the environment

Examples in prior research which fall into this category include research by Navarro, et al., where a KR5arc (KUKA) robotic manipulator and a Schunk SDH2 gripper dynamically interacts with previously unknown objects by tracking, synchronizing to the objects motion and using template matching to leverage experience from previously grasped objects [16]. Research by Luo et al. developed eye-in-hand vision sensing on a 7 DoF robotic manipulator to achieve tracking and fetching tasks for a known object [13]. Finally, Y. Suzuki, et al. demonstrated a system which used a proximity sensor embedded in the palm of the gripper to adjust the grasp to arbitrary perturbations in the trajectory of a spherical object on the conveyor belt [32].

2.4.2 Hitting or batting an object

Robots playing sports is the most common application which requires hitting or batting encountered in the literature. This includes playing table tennis [157, 158, 161, 162], baseball [159], ice hockey [156] and air hockey [163]. These applications present a more unstructured environment relative to the conveyor belt examples discussed above. The object motion is less constrained and the velocity is higher. This greatly increases the demands on sensing, computation and actuation systems which have significantly less time to detect, model, predict, and intercept the object. Despite these difficulties, these applications require a relatively simple interaction with the object, simply hitting or batting it. Since these examples are often presented in the form of a game or sport the robot can leverage a semi structured environment due to the constraints of the game, for example in a game of table tennis the ball will bounce once on the robot's side of the table before it must be hit back, limiting the potential ball trajectories. Furthermore, the objects in these examples are typically very simple. The most common examples are a ball, like a ping pong ball [157], or a disk, like an ice hockey puck [156]. Both of which have the advantage of presenting the same problem regardless of pose and moves in easily predictable trajectories.

2.4.3 Capturing a ball

This group of examples are often a simplification of the task of grasping a dynamic ball. This simplification is achieved by ignoring the motion of the gripper and focusing on some other research question. For example it is common to assume a successful grasp if the end effector comes within an arbitrarily defined range of the path of the object [5, 30, 164]. Alternatively, the problem can be simplified by replacing the role of the gripper with objects like baskets [12, 15], foam rims [17], cups [11, 18], bowls [19], nets [8, 9], magnets [165], and kitchen funnels [23]. This greatly increases the acceptable margin of error in both the estimate of the point of interception and the necessary precision of the actuation system.

2.4.4 Grasping a ball

These examples are similar to the previous two categories in that the test object is a ball, however they use a traditional robotic gripper which is actuated during the interception to attempt to secure the ball. A particularly notable example from the literature uses a 12 DOF, four fingered, DLR-Hand-II gripper, attached to a 7 DOF, DLR-LWR-III arm to intercept and grasp a ball which has been thrown by a human [14]. This research was later expanded to use two instances of the same hardware attached to the Rollin Justin platform to simultaneously catch two balls, one in each hand, using only on-board sensing [6]. Several examples from prior research also examined if a robot can catch a faster moving object by matching the objects speed to reduce the relative velocity between the two [30, 166]. One instance in which this technique was used, a seven DoF KUKA LBR IIWA robot arm, equipped with a sixteen DoF Allegro hand was able to grasp a flying object [22].

2.4.5 Grasping an irregular object

The final grouping which emerged by mapping examples in this way is catching irregular objects. In comparison to the groups previous discussed this represents the most difficult task for a robot to achieve. Since the object is irregular, not only is the trajectory taken by the object harder to model and predict but the robot must model and predict the changing pose of the target object, identify a time in the future at which the object has a position within the manipulators graspable range and a pose which is suitable for an attempted grasp.

One particularly relevant example from the literature presents a robot and grasping strategy capable of grasping irregular objects, both in shape and weight distribution, which are tossed to it by a human. In this example a programming-by-demonstration approach is used and a methodology proposed which probabilistically searches for and finds feasible catching configurations [7]. A possible further complication is grasping soft, delicate or deformable objects. There are many applications where excessive forces applied

to the object, common when grasping moving objects due to the need to rapidly close the gripper, might damage the target object. Prior research has examined how to grasp soft objects while minimising deformation [21]. In this example they used a combination of tactile and vision sensing to adapt the grasping motion in order to minimise contact forces, allowing the robot to autonomously grasp a dropped marshmallow and a paper balloon without damaging either. This focus on grasping fragile objects so as not to damage them is not unique and research has also explored how a robotic system might catch raw eggs [167]. Finally adaptation of the grasping motion of a gripper to tackle spatial errors has also been examined [20]. In that research, a gripper grasped a falling object using a vision system to adapt the grasping motion so as to deflect the object to a more favorable grasping position.

2.4.6 Limitations of current approaches

The above summary and discussion of previous examples of robots autonomously interacting with dynamic objects represents, to the best knowledge of the author, the most comprehensive review of its type to date. Analysing the field in this way has highlighted several shortcomings of prior research.

There is a clear tendency to conduct research in an application-driven manner. The vast majority of the examples given above begin with the goal of developing a robotic platform which can achieve a set goal, i.e. play catch with park visitors [17], play table tennis [157], catch irregular objects [7], etc. This application driven approach leads to methods and findings which work for niche applications but are often limited in their contribution to the field as a whole. There is a need for research which systematically examines the challenges inherently associated with grasping moving objects.

The effect of the environmental conditions is often overlooked. It is very common to demonstrate a robot's abilities in a lab, and to underestimate the advantages that those optimum conditions provide. Lab conditions offer an existing infrastructure of sensors in the environment, often in the form of an advanced motion capture system, lighting

conditions which are optimal, and eliminates the possibility of occlusion. The result of this is that there is a lack of research addressing the issues which arise when interacting with moving objects under real-world, non-optimum conditions.

2.5 Machine learning techniques in robotics

Machine learning refers to a collection of techniques in which a computer uses sample data to learn how to act, as opposed to following explicit rules as defined by a human programmer. Since its initial conception in the 1950s, machine learning techniques have become increasingly effective, particularly in recent decades as many of the initial barriers to its success are overcome. Namely, access to massive amounts of training data due to the emergence of the internet and the additional complexity of the models made possible by access to larger amounts of computing power.

Machine learning has long been an area of interest in robotics [168]. A complete review of robotics literature which discusses machine learning techniques would be vast and is outside the scope of this thesis. For a more holistic review of machine learning there are several review papers available. These are typically divided into literature focused on specific techniques or applications, such as deep learning [169], robotic grasping [170], autonomous driving [171], and multi-agent systems [172] to name a few.

Machine learning techniques are particularly popular for their ability to perform effectively in unstructured environments. A well-trained, machine learning model will generalise and adapt well to situations which are outside of its training data. This is not the case for explicitly programmed algorithms which can only follow explicit instructions and can act unexpectedly in new situations. In the field of robotics, machine learning has been applied to a wide range of applications. Examples include perception in soft robotics [173], human emotion recognition [174], fruit recognition [175], playing table tennis [176] and even underwater robotics [177]. In robotic grasping, it has been used in pose estimation [2], grasp planning [178], grasp stability estimation [153], in-hand manipulation [75, 179] and for tackling occlusion [4] amongst others. Machine learning techniques are also

commonly applied to tactile sensor data, it has been used for tactile servoing [105], tactile surface perception [96, 97], grasp stability assessment using tactile sensors [107, 153], and tactile edge following [110]. It has also been used to identify objects from the tactile data collected while grasping [76, 77, 95].

The remainder of this section discusses machine learning under two headings and aims to give the reader a background in machine learning techniques and give more specific examples of how they have been leveraged in prior research. First, models and agents are discussed. These are the mechanisms which when correctly trained will map inputs to appropriate outputs. Next, learning is addressed. This is the process which uses experience to train a model how it should act.

2.5.1 Models and agents

The first aspect of machine learning which we will discuss are models and agents. These are mechanisms which take inputs and are trained to map them to desirable outputs. These are more commonly referred to as models though they are often called agents in reinforcement learning applications. There are a vast range of these, each with its own advantages and disadvantages for specific applications.

Traditional machine learning models

Random forest is a decision tree technique which uses labeled datasets to 'learn' a series of 'if then' style statements which can be used for classification and regression tasks. This technique was used to determine the robustness of a parallel-jaw grasp using a tactile information at two contact points [180]. In this example there was a massive (7500x) speed improvement over the more traditional Monte Carlo technique used previously for this application.

The K-nearest neighbour (kNN) and support vector machines (SVM) techniques have been successfully used to enable a robot to grasp a moving object. They are used to make an initial guess for the optimised robotic trajectory parameters, taking the initial velocity

of the ball as an input, and enabling online optimisation of the robot motion [30]. kNN is a technique which compares an unseen input to a dataset of labeled examples. The output is determined from which data points the input is 'closest to' in the feature space [9]. SVM is a linear classifier which aims to find a hyperplane in the feature space which maximises the separation between different classes.

There are many other options for models which have not been covered, including hidden Markov models, logistic regression, and AdaBoost [117].

Artificial neural networks

An ANN is a concept inspired by the function of biological neurons in the brain. Made up of a collection of interconnected nodes or 'neurons', an ANN takes an input, passes it through one or more layers of neurons to determine an output. Individually, each neuron takes one or more input and applies some simple function to produce an output. Using this technique, a large number of connected neurons allows a neural network to approximate very complex functions. Each neuron input has an associated weight value. The behaviour of the neural network is 'trained' to the desirable behaviour by modifying these weights.

There are many different types of neural networks, this section will discuss the most common, and give examples of when they were used in robotic grasping. The most basic is a Feed Forward Neural Network (FFNN). In this case a set of input neurons feed data forward, either directly to an output through a set of weights or to a hidden layer of neurons which transform the information further before passing it to the output. This technique was able to predict the initial position and velocity of a ball thrown by a human by analysing the throwers motion [164].

The next type of ANN which is discussed are deep neural networks (DNN). An ANN is considered 'deep' when it has more than two layers of neurons, see Figure 2.10. The increased depth of the network massively increase its ability to model complexity. This has been used to great effect in many applications including in-hand manipulation where an

ANN used tactile sensing from a TWENDY-ONE robot (see Figure 2.4e) to accomplish in-hand manipulation, and was shown to be capable of adapting to previously unseen objects [75]. ANNs have also been used to achieve object recognition based on tactile sensor data during a power grasp. [76] and to infer the surface properties of an object [97].

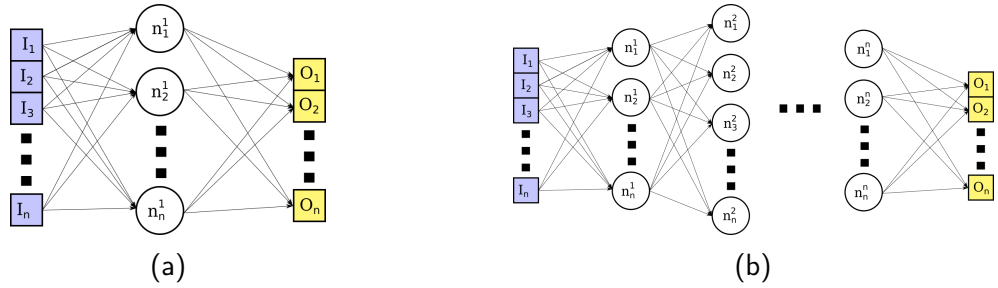


Figure 2.10: Illustration of the difference between **(a)** shallow neural networks (SNN) and **(b)** deep neural networks (DNN)

Particularly popular in image processing applications are convolutional neural networks (CNN). Due to the complex nature of the image recognition problem, a model with a large learning capacity is necessary in order to be able to identify features in images. The success of CNNs when tackling image processing problems has been attributed to the structure of the network makes “strong and mostly correct” [181] assumptions about the nature of images. Furthermore, it has a lower number of neuron connections relative to other network architectures of a similar size making it considerable easier to train. In the sphere of robotic grasping, CNNs have been used to process dense tactile information, leveraging the same relative position information which makes them so effective for processing images. This has been demonstrated to be effective in reconstructing object shape by touch using the GelSlim optical tactile sensor [95], and the uSkin tactile sensor applied to the Allegro robot hand [77] (see Figure 2.7).

There are several other types of ANNs, which are not discussed here, which have been applied to robotics including recurrent neural networks (RNN) [85, 173], and spiking neural network (SNN) [55].

The performance of a neural network can be highly dependant, not only upon the type of network you use, i.e. CNN vs DNN vs RNN, but also network parameters such as depth,

number of neurons per layer, activation functions between layers, etc. For this reason there are number of predefined network architectures which have been found to be effective and are now commonly used for a variety of tasks. These include AlexNet [181, 182], and ResNets [4, 183], LeNet [184], VGG [185], and Inception [186].

2.5.2 Learning

Next we will discuss learning. Learning is a technique which is used to train a model or agent (see Section 2.5.1) such that it performs the necessary transformation to its inputs in order to achieve the desired outputs. There are a variety of methods which are used to achieve this, the most common and relevant will be outlined in this section.

Supervised learning

Supervised learning (SL) is one of the most common forms of machine learning and relies on prelabeled input-output pairs. A common example used is training a neural network for image recognition of hand-written digits. To achieve this a prelabeled dataset is necessary, which includes images of handwritten digits, and a categorical label associated with each image indicating which digit from 0-9 is in the image. This is an example of such as dataset [187]. Supervised learning inputs the image to the model, observes the output and compares it to the label, which acts as a ground truth. The difference between the model output and target output is used to determine how the model can be modified. The goal of which is to minimise the difference between the model output and the target output. This process, repeated many times can train a model to conduct the desired transformation from input vector to output vector, in this example from image to category.

In the context of robotic grasping, SL is a popular tool. One example from the prior literature uses SL to train a model, an SVM classifier, using synthetically generated data to estimate grasp stability from tactile sensors on the Barrett robotic hand [107]. Another example involved training a random forest model and a deep neural network to estimate the robustness of a parallel-jaw grasp [180].

Unsupervised learning

Unsupervised learning does not use labeled data sets, instead it is used to examine and extract relationships in data. There is no target value or category, instead unsupervised learning performs clustering and density analyses of features in the feature space. In the context of robotics this increases autonomy by removing the need to label data and allows the robot to learn through experience in the absence of human intervention. This has been applied to the problem of autonomous tactile surface perception whereby an agent was able to classify, with very high accuracy, a collection of different surfaces without the need for labeling or specifying the number of categories [96].

Reinforcement learning

A technique which has been gaining in popularity in recent years is reinforcement learning (RL). RL is used to train a model, sometimes called an agent in RL, through trial and error. Markov decision processes are the basis of RL techniques, formally described as a tuple (S, A, P, R) , where S and A represent the state and action space respectively. P represents the state transition probability model, i.e. how likely the system is to transition from one state to another. Finally R represents the reward function, this the mechanism by which specific behaviours are encouraged. The agent explores its environment by observing some state S , samples an action from the possible actions A , and then the environment moves to a new state according to P . The agent receives some reward R associated with the sampled action, positive reward for desirable behaviours, i.e. winning the game, scoring a point or grasping the ball, and zero or negative reward otherwise. The goal of the agent is to learn some policy $\pi(S_T, A_T)$ which maps observations about the state to actions so as to maximise the accumulated rewards. In this way the agent learns by exploring the environment, i.e. sampling different actions, and having its performance evaluated via the reward mechanism.

A number of examples in the prior art have demonstrated the effectiveness of RL techniques in robotic grasping. This includes research which used Q-learning to train the

Baxter robotic platform in the Gazebo simulation environment. In this research the robot successfully learned a grasp policy which increased grasping accuracy [188]. Furthermore, research by Hoof et al. shows the ReFlex robot hand learning in-hand manipulation skills which generalise to previously unseen objects [179].

Policy gradients

A particularly popular technique used in RL is policy gradients (PG). In PG, the policy is parameterised and a gradient vector calculated to update the parameters so as to maximise the reward. In this technique the agent operates in an environment, takes inputs and produces an output which is mapped to an action. For each input-output pair, an action is sampled (pseudo-randomly as opposed to greedily) and executed. Unlike Q-learning, where the value function or quality function is used to estimate and assign the expected reward, the future rewards which will be attributed to the sampled action is not yet known. Instead the episode is run until the end and outcome observed. It is at this point where the rewards associated with each action can be calculated and rewarded. A gradient vector is calculated enabling each parameter, i.e. weight in a neural network, to be modified such that positive rewards will be more likely in the future. The agent can be trained by repeating this process for a large number of episodes, giving positive rewards when the sampled action produces the desired behaviour and negative rewards when it does not.

Deep learning

Deep learning is not an alternative to the examples of learning given above, but instead refers to the techniques which are used in addition to the above techniques to train deep neural networks. Whereas supervised, unsupervised or reinforcement learning techniques determine how a neural network is trained, deep learning provides the tools with which to manipulate the network weights so as to produce the desired effect. Deep learning has shown state-of-the-art performance when compared to traditional machine learning approaches in many machine learning applications including, computer vision [175],

speech recognition [189], robotics and control [190], among others [169].

An essential aspect of deep learning is back-propagation. Back-propagation is an algorithm which is used to manipulate a neural network so as to encourage or discourage a particular behaviour. It does this by calculating the gradient vector, i.e. the effect each weight in the network has on the output, which can then be used to train the network. For example in supervised learning, an error is calculated by comparing the network output to the target output and this error minimised by modifying each weight in the direction which will reduce the error. The final goal being, to have the weights converge onto values which minimise the error.

Transfer learning

Transfer learning refers to taking a model which has already been trained in one environment, application or dataset and using some or all of it as a starting point for training a model or agent for a related environment, application or dataset. This can be a very powerful technique and significantly reduce the amount of training data and training cycles required. Examples of this can be seen in robotics where transfer learning was applied to a model trained for material classification to create a model for the related task of haptic classification, massively reducing the cost of creating a haptic classification model from scratch [97].

Synthetic data

Synthetic data is a common way of combating the need for large amounts of training data [2, 153]. Synthetic data refers to data which is artificially created rather than collected from real world events. For example it can be generated from simulation or extrapolated from existing data. It is a powerful tool, enabling training to be conducted for applications which would otherwise be very costly or potentially impossible. It is also prone to several common errors. Synthetic data can be a good approximation of real world data but is often no substitute for the real thing. Furthermore synthetic data can often lead to overfitting, where by the trained model performs well on the data generated

by the model but then struggles when the model is implemented in the real world. There are many examples of synthetic data being used in robotics, including data generated for pose estimation during grasping [2].

Existing robotic grasping datasets

One tool which can be used to tackle the need for huge amounts of training data is to use existing datasets. These are enormous collections of data points for the purpose of machine learning applications and are often shared publicly for the research community to use. The advantage of these is the easy access to large amounts of training data, however they are limited to the data type which has been collected as well as the categories assigned, the labels applied, and the chosen format. These may not perfectly align with the target application.

The most well-known datasets like MNIST [187] and ImageNet [191] are typically for image processing applications, however datasets also exist for robotic grasping applications. Some include a mix of synthetic and real world data such as Dex-Net [192] while others like the Google Grasp Dataset [190] conducted 800,000 grasping attempts in the real world, running between six and fourteen robots over the course of two months, this can be seen in Figure 2.11. Dex-Net was designed to estimate the probability of success for a parallel plate gripper grasp while the Google Grasp Dataset recorded monocular camera images during grasps to infer information about the spatial relationship between gripper and camera. This type of data generation is not unique [182], but is incredibly costly and difficult to conduct for every application. Another dataset relevant to robotic grasping is the Cornell Grasping Dataset, which contains 885 images of 240 graspable objects each labeled with a graspable rectangle. This allows the training of parallel plate grippers to identify potential strategies of grasping an object [193]. A dataset which includes both tactile and vision sensor data is the THU dataset, this was collected to assess grasp stability with the Barrett hand [194]. One of the issues with relying on existing datasets for training, that the data available may not be useful for your application, is particularly apparent when examining the datasets available in the robotic

grasping space. There is a large amount of data for pincer and suction grippers, however there is a lack of data for grippers of other morphologies. Due to the limitations of using existing datasets, simulation is often used to generate sufficient data and is discussed in the following section



Figure 2.11: Large scale data collection setup used to create the Google Grasp Dataset [190]. 14 robots collected 800,000 grasp attempts

2.6 Robotic grasping in simulation

Simulation has long been a tool used in robotics, its popularity can be attributed to the fact that it is faster and less resource intensive than real-world testing with a physical robot [157, 164, 166]. The applications of simulation to the field of robotics can broadly be categorised into three categories, planning, learning and experimentation.

The use of a simulation as a planning step refers to a robotic system using its current model of the world to simulate a set of proposed actions to attempt to predict the outcome [195]. By predicting a likely outcome in this way the robot can make an informed decision about how to proceed.

Simulation as a tool for machine learning has become increasingly popular in recent years

[196]. With the rising popularity and potential of machine learning techniques in robotics, the problem of the lack of training data [153, 197] and the difficulty of running large amounts of training cycles in the real world came to the fore [188, 198]. A common response to this issue is the use of simulation, both to generate synthetic data [2, 30] and to enable off-line, simulated training which can achieve a large number of training cycles with relative ease [188]. An example of this is the robotic platform Baxter which used reinforcement learning, executed both in simulation and in the real-world, to learn a grasping policy [188], this is shown in Figure 2.12.

It is also common to create simulations to test hypothesis at significantly lower expense and time commitment than would be required to achieve the same experiment in the real world [3, 30, 82, 92, 153, 188, 199, 200]. An example of this is the simulated testing conducted on the iCub robotic platform which used simulation to first learn the dynamic behaviour of irregularly shaped objects and then tasked a simulated iCub platform with catching a hammer and a tennis racket [7]. This simulation is shown in Figure 2.13.

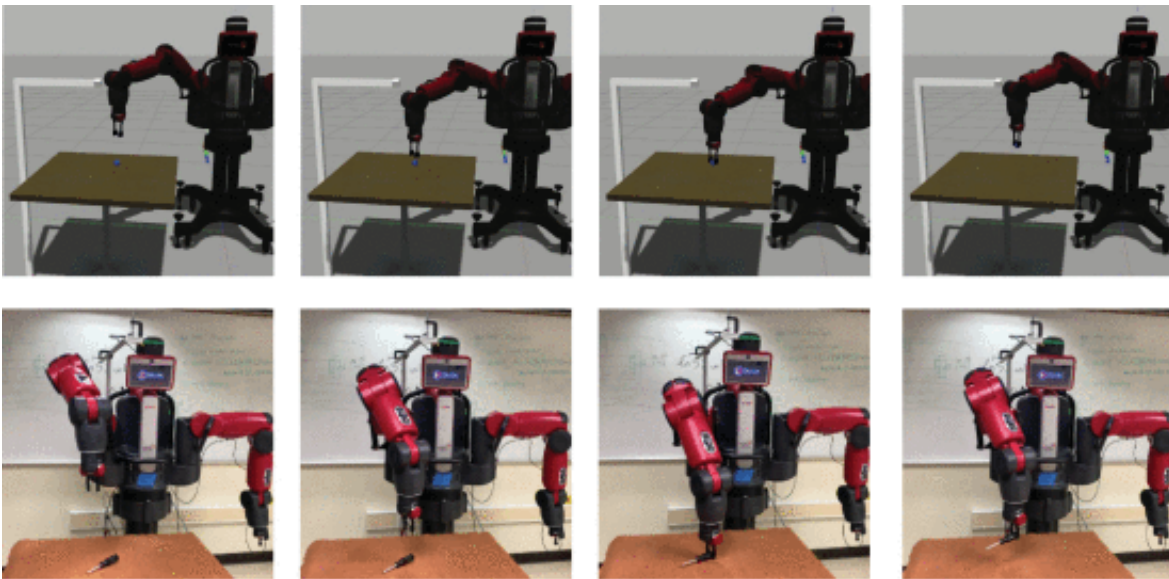


Figure 2.12: A simulated and real-world Baxter robot learning to grasp using reinforcement learning. The simulation is conducted using the Gazebo robotic simulator

Though popular, simulation is inherently limited by how accurate it can be made.

Furthermore, when attempting to achieve greater levels of accuracy in simulation it is often a case of receiving diminishing returns on investment. Commonly cited problems with simulation include, but are not limited to, object deformation [1, 86, 201], cluttered

environments resulting in multi-contact systems [199], and a “reality gap” between simulated and real-world sensor data [2]. Although there are common methods used to tackle these shortcomings, for example domain randomisation [197], artificially induced sensor noise [82], etc., it is essential to understand the limitations of simulation accuracy. For this reason it is common to conduct simulated and real world testing in parallel [86, 165, 188, 198, 199, 200]. In this way the advantages of simulation can be leveraged while also using the real world experiments to examine the accuracy of the simulation and verify the results.

There are a wide range of simulation tools available for robotic applications. The most commonly used physics engines used are the Open Dynamics Engine (ODE), Bullet, MuJoCo [202], Newton and Vortex. Each of these operate in slightly different ways and the context dictates the most appropriate tool to use. While it is possible to develop robotics simulations using these physics engines directly [199] or even without a physics engine by simulating the relevant contact and constraints manually [200], most researchers choose to reduce the development difficulty by using a robotics simulator or framework. These provide tools such as visualisation, robotic models, common robotics algorithm implementations (such as forwards and inverse kinematics) and commonly used sensors like tactile and image sensing. This massively reduces the skill level and time required to produce useful simulations. These include V-Rep [203], PyRep [204], Gazebo [205], RobWorkSim [201], GrasplT [206] and Webots [207]. These simulators typically vary slightly by specialising in a particular area of robotics research. Some, like Gazebo, are general purpose while others focus on robotic learning or grasping and manipulation, like PyRep and GrasplT respectively.

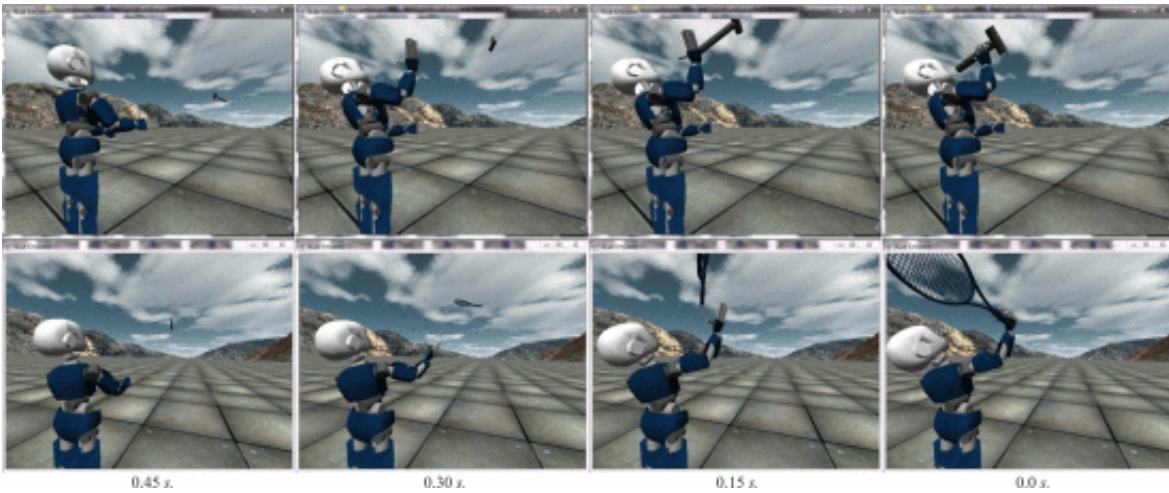


Figure 2.13: Robotics simulator evaluating iCub’s ability to grasp a moving object

2.7 Summary

Research on robotic grasping of moving objects dates back to the 1990s [154] and there are numerous examples of robots interacting with moving objects. Examples include picking objects from a conveyor belt, batting, juggling, grasping, and catching. An application-centric focus in prior research has led to a lack of research which uses systematic techniques to examine issues inherent to the problem of grasping dynamic objects. This thesis proposes an experimental methodology which defines the grasping conditions in terms of distinct parameters, thereby allowing a systematic examination of the performance of different grasping strategies across a range of grasping conditions.

Prior research has neglected to address the effect of real world grasping conditions, instead conducting demonstrations in ideal, laboratory conditions. Many of the examples discussed rely on highly accurate interceptions of the moving object by the robot. The level of accuracy required to achieve a successful interaction is often difficult or impossible to achieve under real-world conditions. There is need to conduct foundational research which examines the effect of non-optimum interceptions caused by real-world conditions, and to create strategies which mitigate the negative effects of a non-optimum interception. This thesis conducts testing on a range of control strategies while

systematically inducing interception errors in a controlled and repeatably way. In this way, a strategy's suitability for deployment under real-world conditions can be examined by quantifying its robustness to errors which are common under these conditions.

Tracking the far-field trajectory of an object is a fundamental part of autonomously interacting with a dynamic object. To date, this has been achieved by using an arrangement of cameras in the area surrounding the robot. This approach must contend with the limitations of relying solely on vision systems. Sensing the near-field position of the object relative to the gripper has received little attention in the literature. Despite the effectiveness of tactile sensing in other applications, and in particular in unstructured environments, the potential contribution of tactile sensing to track the near-field position of the object while grasping moving objects remains under explored. This thesis proposes, implements and tests a reactive control strategy which leverages real-time tactile sensor data to create adaptive grasping motions which attempt to mitigate the negative effects of grasping under real-world conditions.

Figure 1.1 shows an overview of the process of interacting with a dynamic objects, broken into five steps. There is a lack of existing research which focuses specifically on optimising the grasping aspect of the fifth step, intercepting the object. This deals with the role played by the gripper while grasping moving objects, which has to date been under estimated and under researched. The majority of examples simplify or eliminate the role of the gripper by using baskets, nets or other devices. The small subset of research which do accomplish grasping of a dynamic object with a traditional robotic gripper use predefined gripper closing motions and extensive parameter tuning. The potential contribution of a grasping motion which can adapt to the position of the ball in real-time, during the grasp, is yet unexplored. This thesis demonstrates two controllers which include adaptive grasping motions, the first adapts by following a set of heuristics and the second learns an adaptive motion using machine learning techniques.

Machine learning has been shown to a highly effective tool in robotic grasping. It is particularly attractive for its ability to adapt to previously unseen circumstances making it

an effective solution in unstructured environments. This suggests that machine learning techniques could be highly suitable for the task of grasping moving objects and in particular as a tool to leverage tactile sensor feedback and realise the adaptive grasping motion discussed previously. This thesis proposes, trains and deploys a control strategy which uses a neural network trained using machine learning techniques. The performance of this strategy is quantified across a range of grasping conditions and compared to the performance of a heuristic-based, reactive control strategy and a traditional, predictive control strategy.

3 | Methods

The limitations of existing research have been discussed in previous chapters. The need to explore strategies which leverage alternative types of sensing and adaptive grasping motions has emerged. Any proposed strategy which leverages these techniques will need to be systematically tested and its performance quantified and compared to a traditional control strategy. This chapter aims to develop a methodology which can quantify and compare the performance of different control strategies under a range of grasping conditions. This methodology is broken into three categories. First the experimental methodology is addressed. Next, the statistical methods used are outlined, enabling an objective analysis of the results obtained. Finally, the control strategies are outlined. These are a traditional control strategy common in the prior art and a novel, reactive control strategy.

3.1 Experimental methodology

This experimental methodology has several key objectives. The methodology should:

- **Quantify the performance.** To quantify the performance of a gripper while grasping a moving object, the gripper must attempt a grasp, and each attempt deemed a successful or failed grasp. This methodology must define a success criterion to determine the outcome of each attempted grasp.
- **Compare control strategies.** The performance of different strategies should be isolated, quantified, and compared. This methodology must ensure that the differences in observed performance are solely due to the control strategy used, and

not impacted or limited in some other way.

- **Test under a range of grasping conditions.** The suitability of a grasping strategy for deployment in an unstructured environment can be better understood by systematically quantifying its performance under a range of grasping conditions. This experimental methodology must identify key parameters of the grasping conditions and include mechanisms which vary these in a repeatable way.

To address these requirements, an experiment was formulated where a robotic gripper repeatedly attempts to grasping a rolling ball. The ball is moving toward the gripper, along a horizontal plane, where the position of initial gripper-ball contact, the timing of grasp initiation and the object speed is systematically varied from test to test. These key parameters of the grasping conditions are discussed further, later in this section. A successful grasp is defined as one which can support the entire weight of the object, in the absence of external forces, for a time period of five seconds.

By conducting multiple tests, each with a binary outcome, a grasping success rate which reflects the performance of the gripper can be calculated from Equation 3.1. Each strategy is implemented using the same computational resources, sensor interface and actuator mechanisms. This ensures that the computational power and control mechanisms available to each strategy is consistent and does not advantage one strategy over another. Finally, by systematically varying key parameters of the grasping conditions, and conducting multiple tests for each set of conditions the performance of each strategy under a range of grasping conditions can be quantified.

$$\text{grasping success rate} = \frac{\text{number of successful grasps}}{\text{number of attempted grasps}} \quad (3.1)$$

A key aspect of this methodology is varying the key parameters of the grasping conditions. Existing literature was used to identify key factors which affect the success of grasping moving objects under real-world conditions. These were used to describe the grasping conditions as three distinct parameters. Traditional strategies typically rely on estimating an appropriate point for the robotic gripper to intercept the object and

attempting to move the gripper into that position in time to grasp the object.

Imperfections in sensing, processing and execution, will result in some interception errors between the attempted grasp and the optimum grasp. These can be represented under two categories; a) as a positional error, i.e. an offset between the center of the gripper and the position of initial contact between gripper and object (see Figure 3.1), and b) as a temporal error, i.e. the time difference between the optimal time to initiate the grasp and the actual time the grasp is initiated. A positional error and temporal error are the first two parameters used to describe the grasping conditions while the final parameter is the speed of the target object.

This experimental methodology provides a highly repeatably way to address the requirements identified, while keeping the required apparatus simple and inexpensive. The subsections below isolate and discuss key components of this apparatus.

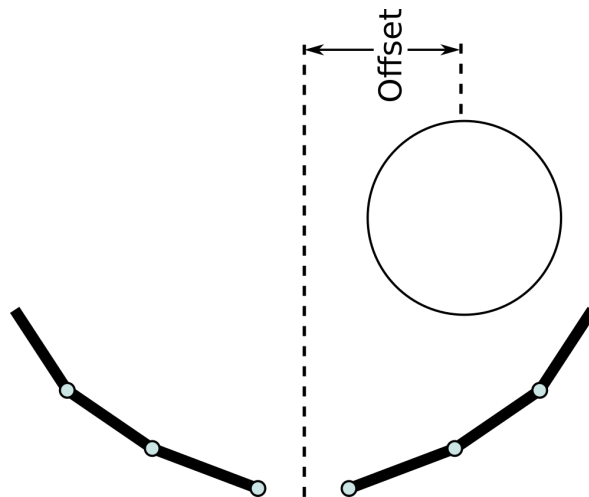


Figure 3.1: Offset between gripper-object contact point and gripper centre, representing positional error in the interception of a moving object by a gripper

3.1.1 Gripper morphology chosen for testing

It was important that the results and associated conclusions can be extrapolated to a wide range of robotic grippers. The test gripper should not benefit from features only present on a small subset of grippers. Guided by this, the kinematics of individual fingers are based on the kinematics of a human finger. The popularity of bio-inspired designs in existing robotics grippers ensures that results are applicable to a wide range of grippers.

Furthermore, the chosen gripper morphology should: be capable of grasping a moving ball using a power grasp, maintain the simplicity of the experiment and maximise repeatability.

Therefore, a two-finger gripper, with fingers in opposition to each other, is chosen as the gripper's morphology.. Each finger should have three DoF and similar dimensions to a human finger. At this point, it is appropriate to define some terminology which is used when discussing the gripper. The left finger is defined as the finger which when viewed from behind the gripper, i.e. the gripper between the observer and the object, is on the left (see Figure 3.2). Furthermore, terminology used when discussing the human finger is also commonly used when discussing robotic fingers with similar characteristics. This research continues this practice and refers to each phalanx as the proximal, middle and distal phalanges and the joints as the metacarpal phalangeal joint (MP), the proximal interphalangeal joint (PIP) and distal interphalangeal joint (DIP), in order from gripper center to fingertip, see Figure 3.2.

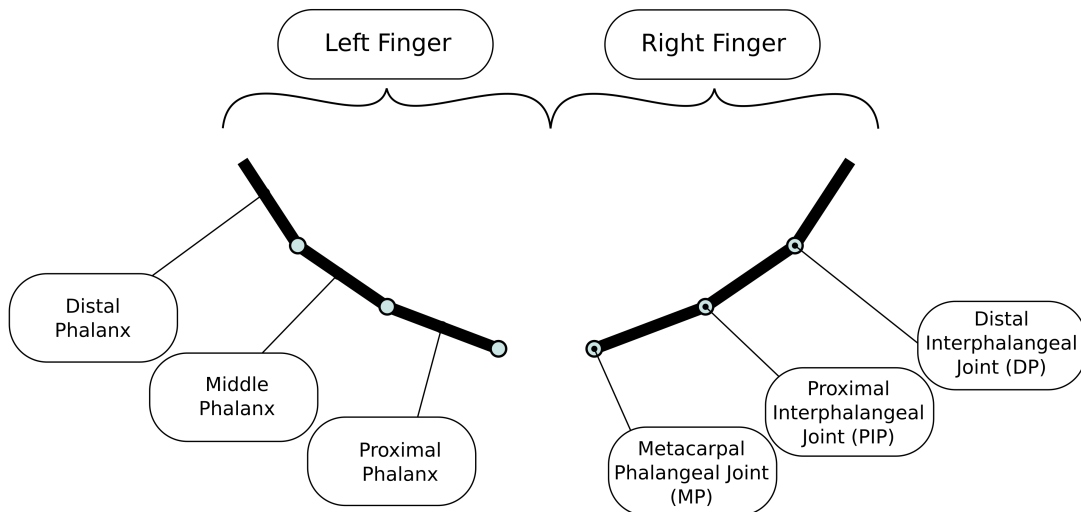


Figure 3.2: Diagram of gripper morphology, including labels which are used to refer to the different fingers, phalanges and joints

3.1.2 Choice of test object

The object shape chosen in this experimental methodology is a sphere. The choice of a spherical object simplifies the grasping problem by eliminating the effect of object

orientation and allowing the experiment to isolate and examine the effects of interception errors, object speed and control strategy. This strategy of eliminating orientation as a variable to focus other factors is common in the prior art [9, 14, 15]. Furthermore, a spherical object has the additional advantage of simplifying the mechanism used to set the object in motion during an experiment. A sphere can be set in motion, its path accurately controlled, and its speed varied, using a simple inclined track.

3.1.3 Positional interception error

A key parameter of the grasping conditions tested is the positional error. This is the distance between the optimum and actual position of initial gripper-object contact.

Existing techniques overlook challenges associated with deploying robotic systems in unstructured environments, thereby limiting the portability of the demonstrated functionality. Positional errors are a critical hurdle to grasping a moving object in an unstructured environment. By isolating and examining the effect of positional interception errors, a better understanding of their impact on grasp success can be achieved. This has the potential to inform the development of strategies which can achieve successful grasps despite the presence of these errors, thereby creating strategies more suitable for deployment in unstructured environments.

Identifying the optimum interception position

The chosen definition of the optimum interception position in this research is the one which is most robust to deviations from that position. This may not necessarily correspond to the point which yields the highest grasping success rate, but is appropriate given the focus of this research on robustness to errors and operation under non-ideal conditions. The symmetrical nature of the gripper strongly suggests that this can be assumed to be the center of the gripper, though this assumption should be verified empirically.

Induce offsets from the optimum position

Testing is then conducted at a range of offsets from this optimum to examine the effect of positional errors in the interception. The simplification of the problem to grasping a sphere on a horizontal plane allows this error to be induced in a highly repeatable and accurate way by simply controlling the relative positions of the track and gripper, illustrated in Figure 3.3.

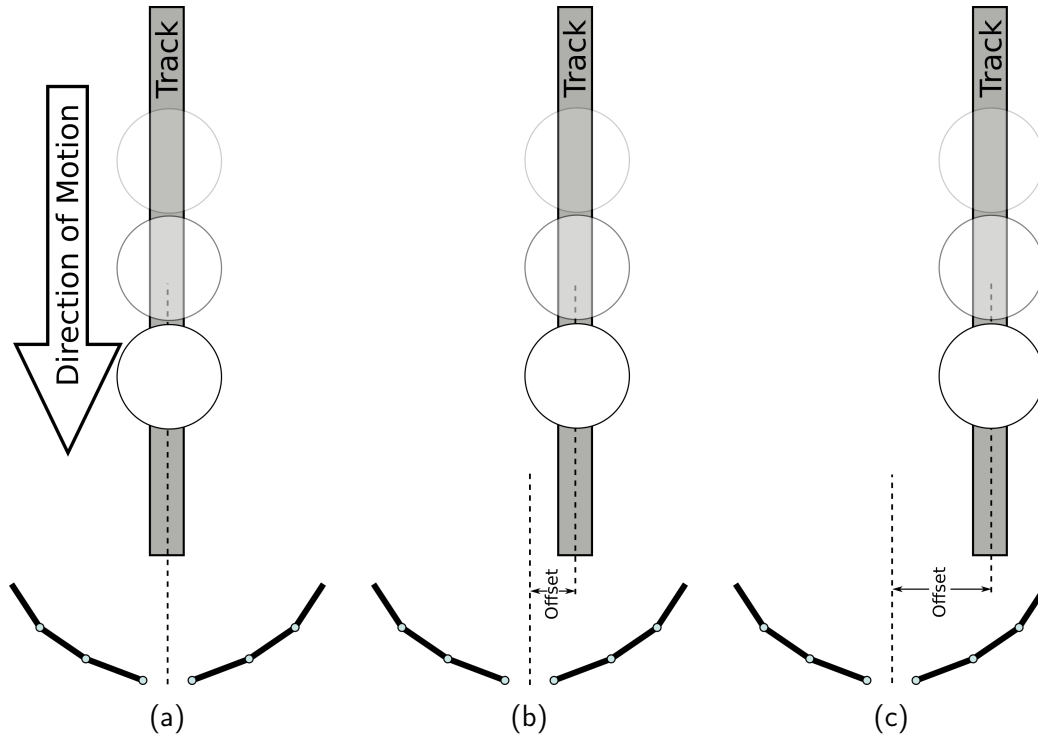


Figure 3.3: Illustration of the positional offset, (a) no induced positional offset, (b) a small induced positional offset (c) a large induced positional offset

3.1.4 Timing of grasp initiation

The second parameter which describes the grasping conditions, and as such is an independent test variable, is the time of grasp initiation.

Prior examples of robotic platforms autonomously grasping dynamic objects rely on manual tuning a grasp initiation time based on the position of the object. This approach enables grippers to achieve successful grasps under a limited range of conditions, i.e. similar object speeds, sizes, etc, but lacks the flexibility to adapt to a wider range of conditions. Isolating and examining the effect of grasp initiation time is essential to better

understand its impact on grasp performance and how to adapt the grasp initiation time to different grasping conditions.

Identifying the optimum grasp initiation time

Unlike the positional error where the geometry of the gripper and 2D nature of the experiment enable an assumption of what the optimum interception position is, the optimum grasp initiation time is much more difficult to identify. Before applying this experimental methodology to a robotic system, the optimum time to initiate the grasp, for that system, must first be determined empirically.

Induce offsets from the optimum grasp initiation time

Offsets from optimum can be introduced to examine the effect of temporal errors in the grasp interception. The simplified nature of the grasp and the highly controlled nature of the path taken by the object allows for highly accurate tracking of the object as it approaches the gripper. This makes testing a range of grasp timings in a repeatable way relatively simple.

3.2 Velocity of the test object

The third and final parameter describing the grasping conditions is the object velocity. The velocity vector of the test object can be broken into two components, the object speed and the angle of incidence. This methodology does not consider the effect of changes to the angle of incidence, instead all testing is conducted with a velocity vector which is 90° to the gripper, as shown in Figure 3.4.

The speed of the moving object is a critical aspect of the grasping conditions. Higher object speeds reduce the size of the window of time which the object is in the gripper's graspable region. Furthermore, higher object speeds have more kinetic energy changing how the object and gripper interact during impact. This includes the deformation experienced by the gripper and the angle of the rebounding object. Previous examples of

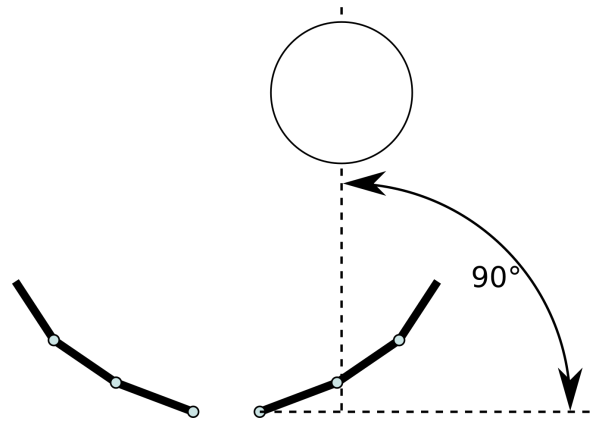


Figure 3.4: Illustration of the angle of incidence, which is constant at 90° for each test

robots grasping moving objects are often application focused. They tend to demonstrate the successful completion of a predefined task by the robot, i.e. a robot grasping an object thrown from a particular range or a robot juggling two balls. The result of this is the robot only has to deal with a small range of object speeds and the control strategy employed can be tuned specifically for those needs. A robot which must operate effectively in dynamic environments must be able to adapt to a larger range of object speeds. Furthermore, this application focused approach lacks a systematic examination of the effect of speed and results in a lack of understanding of the role it plays when grasping moving objects. Testing is therefore conducted at a range of object speeds.

3.3 Statistical analysis

Chi-square tests of independence are used to compare the results of testing on different control strategies. This is chosen as an appropriate technique since chi-squared tests are intended to be used when the variables being studied are categorical. This application has two categorical variables, each attempted grasp has an associated control strategy, i.e. predictive or reactive, and an associated outcome, i.e. successful grasp or failed grasp.

Chi-squared tests of independence estimate the likelihood that two categorical variables are significantly related or independent and is based on comparing the expected and observed frequencies for a particular categorical variable. In the context of the testing

conducted during this research, the null hypothesis is that there is no relationship between control strategy and grasping success rate. Assuming the null hypothesis is true, the expected frequency of successful grasps for each strategy is calculated. This is then compared to the observed frequency to determine if the null hypothesis should be rejected or not. Thus indicating, if there is a relationship between control strategy and grasping success rate. An aspect of the chi-square test worth noting is that it only examines independence and does not indicate the direction of the relationship, i.e. if the control strategy had a positive or negative impact on performance. If the chi-square test rejects the null hypothesis then the observed grasping success rates can be used to determine the direction of the effect of the control strategy.

The statistical analysis employed in this thesis uses a version of the chi-squared test which includes Yate's correction for continuity. This correction technique aims to compensate for a false assumption in the original form of the chi-squared test, that the expected frequencies calculated during the chi-squared test can be represented as a continuous distribution. This correction is commonly used in situations where the expected frequency can be lower than five. This technique comes with its own drawbacks and can tend to over-correct. That said, the error associated with this version of the chi-square test tends to cause larger p-values, resulting in a conservative conclusion, i.e. concluding that the control strategy and grasping success rate are not significantly related when in fact they might be.

3.4 Strategies for grasping moving objects

Several control strategies for grasping moving objects are developed and compared. First, a control strategy which is representative of approaches seen in the literature is developed and deployed on a robotic gripper. This acts as a benchmark to which the performance of alternative strategies are compared. Next, a reactive strategy which is capable of adapting its grasping motion based on real-time tactile sensor data, is developed and deployed. These control strategies, called the predictive and reactive strategies respectively, are the

focus of the initial investigation, and are discussed below. A control strategy which leverages machine learning techniques is also examined and is discussed in Chapter 6.

3.4.1 Predictive grasping strategy

The first strategy, a predictive grasping strategy, provides a proxy to current approaches and acts as a control to which the proposed, reactive strategy, and any future strategies tested, can be compared.

There are several reasons a custom strategy which is representative of existing approaches should be developed, rather than employing one verbatim from the literature. A systematic evaluation of the strategy's performance under a range of grasping conditions is only possible using the experimental methodology developed and described in section 3.1. The simplification of the grasping problem to a 2D problem of grasping a rolling sphere enables a systematic performance evaluation but requires the adaption of existing strategies. Furthermore, the control strategies identified in the literature follow a predictive strategy structure but contain slight differences in their specific implementations, i.e. when the grasp is triggered, how the ball trajectory is modeled, etc. These are carefully considered while developing the predictive strategy used in this research. Finally, the application driven method in which the control strategies seen in the literature are developed would limit the ability to generalise the results obtained to a large range of applications and settings. It is essential that the strategy tested during this research represents the inherent strengths and weaknesses of predictive strategies without the nuanced flaws and features of specific implementations. In this way, the performance can be fairly compared to its reactive counterpart's performance and enable an informed discussion comparing predictive and reactive control strategies for grasping moving objects.

The predictive strategy used in this research relies solely on an interception position and the grasp initiation time to inform how it attempts to grasp the object. This is

representative of predictive strategies deployed in the real world which would estimate an appropriate position and time to intercept the object. The strategy must assume that this position and time are optimal, though the experiments will be conducted under a range of induced interception errors. During these tests the initial position of the gripper is the estimated interception position for that test, and as such the predictive strategy will not need to move the gripper. The grasp timing is slightly more complex, the predictive strategy relies on sensors which track the ball as it approaches the gripper and which trigger the grasp at the estimated interception time. At this time the gripper closes both fingers simultaneously in a predefined closing motion. This behaviour is visualised in Figure 3.5.

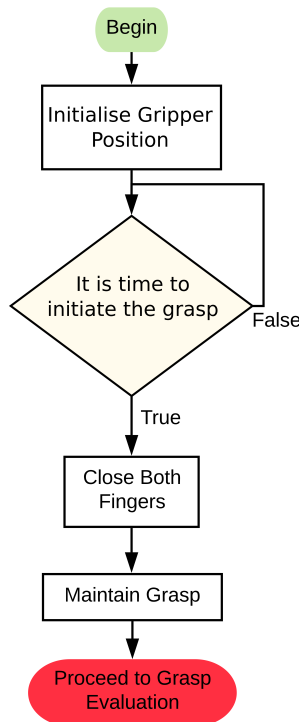


Figure 3.5: Flow chart illustrating the predictive strategy

3.4.2 Reactive grasping strategy

The second control strategy is a novel reactive strategy. In order to address the research hypothesis, the reactive strategy must fulfil two requirements: it should use tactile sensing to detect the position of the ball during the grasp, and it should leverage this sensor data

to create an adaptive grasping motion which attempts to mitigate the negative effects of interception errors.

The proposed strategy is implemented using three basic heuristics:

1. The grasp was triggered when the object first made contact with any of the gripper's tactile sensors.
2. Upon detection of contact with an object, the gripper moved laterally so as to reduce the positional error and centre the object in the gripper.
3. To minimise forces exerted on the ball at contact with the object, the closing motion of the finger that first comes into contact with the ball was delayed, relative to the other finger.

A flow chart demonstrating the implementation of the reactive strategy is shown in Figure 3.6, furthermore an illustration of the gripper executing this strategy is shown in Figure 3.7.

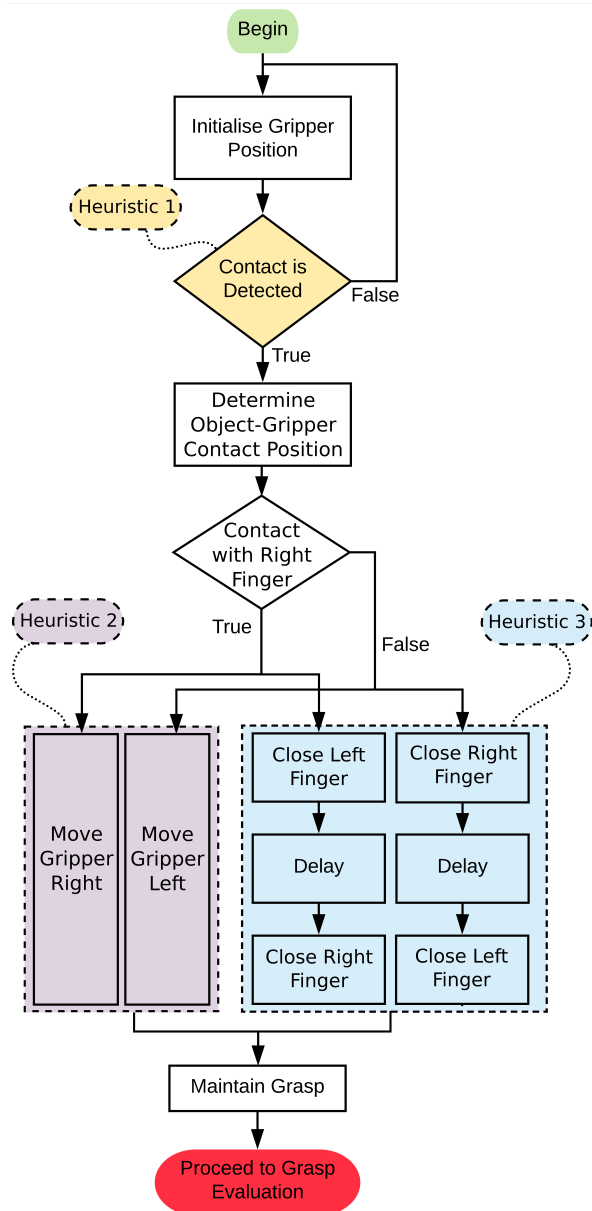


Figure 3.6: Flow chart illustrating the reactive strategy

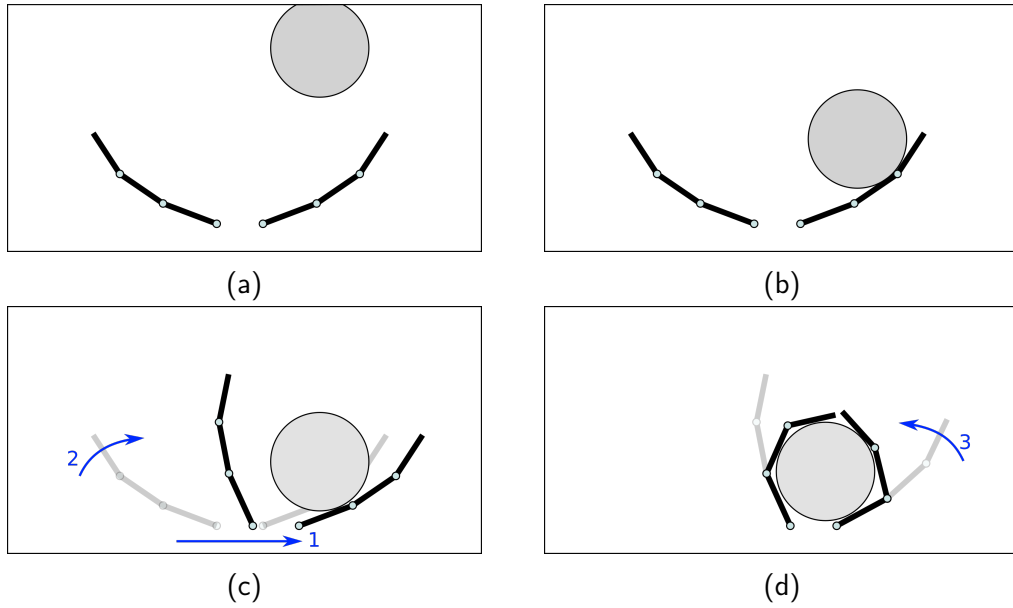


Figure 3.7: Illustration of the reactive grasping strategy: (a) Stage 1: Ball approaches gripper with an induced spatial offset (b) Stage 2: Ball makes contact with the gripper, a tactile sensor in the right finger triggers the grasp (c) Stage 3: Gripper begins to grasp the ball, this involves moving laterally to the right, and closing the left finger (d) Stage 4: Grasp is completed by closing the right finger

3.5 Summary

This chapter proposes a testing methodology which simplifies the task of autonomously grasping a moving object, such that specific parameters of the interception conditions can be isolated and examined independently. In this way, the performance of different grasping strategies can be quantified and compared. This provides a fundamental tool necessary to examine the effect of adaptive grasping motions on grasp robustness.

Furthermore, statistical methods are identified and outlined, which enable the results of testing to be analysed. This is essential to quantify the level of significance seen in the performance improvements as a result of the gripper control strategy.

Finally, two control strategies are outlined. These are: a strategy which is representative of the current approach to this problem, and a novel, reactive strategy which leverages real-time, tactile sensor data to adapt the motion of the gripper during the grasp. The strategies outlined will serve as the initial test strategies.

A key contribution of this research, is the methodology proposed in this chapter. An

examination of the existing literature reveals a heavy reliance on application and demonstration based research, where robotic grasping is applied to specific challenges such as enabling a robot to grasp a tennis racket [7], gently catch a marshmallow [21], or grasping a fast moving ball [22]. The associated literature focuses heavily on the specific implementation used to address the target application. Though these provide important insights into the challenges associated with grasping moving objects, there is a lack of understanding of the fundamentals of the challenge of grasping a moving object. This methodology allows testing to abstract from the application and provides a better understanding the how specific grasping conditions affect grasping success as well as the potential contribution of adaptive grasping motions.

The following chapters will leverage the methods developed here in order to test the primary hypothesis of this thesis, that an adaptive grasping motion, based on real-time tactile sensor feedback, could mitigate the effects of errors in the interception of a ball by a robotic gripper, resulting in a more robust control strategy for grasping moving objects. These methods will first be implemented and utilised in simulation in Chapter 4, then in a real-world setting in Chapter 5. Chapter 6 will utilise the testing methodology and statistical methods to create and evaluate a grasping strategy using machine learning techniques. Finally, Chapter 7 will conclude this thesis, summarising the contributions to existing research and outlining areas of future work.

4 | Validation through Simulation

Having established a suitable experimental methodology and defined predictive and reactive grasping control strategies, this chapter applies these techniques in simulation. This chapter aims to achieve several key goals. First, to develop and calibrate a simulated testing environment which implements the experimental methodology and control strategies, outlined in Chapter 3. Next, to conduct simulated testing to quantify and compare the performance of the predictive and reactive control strategies. Finally, to use the findings of this investigation to inform the direction of future research. To achieve this, a simulation of a robotic gripper, grasping a moving object is developed. This is calibrated by comparing test grasps, executed by the simulated gripper to similar grasps, conducted on a real-world gripper with the same morphology. Finally, each of the control strategies are deployed on the simulated gripper, and testing conducted to determine a grasping success rate for a range of grasping conditions.

4.1 Simulator Development

The simulation developed in this chapter, has several requirements which must be addressed during its design and development. In order to conduct meaningful testing, a simulated gripper embodiment which realistically reflects the geometry and behaviour of a real world robotic gripper is necessary. This includes the need for an accurate simulation of tactile sensing. Accurate simulation of the physics in the environment during testing is also required. In particular relating to the behaviour of the ball while interacting with the gripper. Finally, the simulation must facilitate the methodology outlined in Chapter 3, i.e.

it should allow a quantitative evaluation of grasping performance, the deployment of multiple different grasping strategies and the control of several key parameters of the grasping conditions.

Before discussing the specifics of the simulated gripper and environment which was developed, several key decisions were made regarding the tools which would be used to facilitate this development. These include:

- **Choosing a software framework.** Robotic operating system (ROS) is a modular reconfigurable software framework which promotes code reuse and offers a suite of functionality that can readily be applied to a wide range of robotic platforms. The software is developed in functional modules called "nodes", each responsible for a specific task. Communication between these nodes is facilitated through the ROS framework. This functionality is used to implement the software needed to control the simulation, implement the experimental procedure, implement the gripper control strategies, and automate testing.
- **Choosing a robotics simulator.** There are a range of available simulators which have been applied to solve robotics problems, these have previously been discussed in Chapter 2. These existing tools hugely increase the simulation fidelity and accuracy which is feasible, by removing the need to create these tools from scratch. Several simulator requirements were identified as essential to this research, these include accurate dynamic simulation, 3D graphics, native sensor integration, and ROS compatibility. Though a few of the aforementioned simulators deliver on all or most of these, Gazebo was chosen for this research since it not only delivered on these requirements but also has several other advantages. Gazebo has a large active community which assists with development, it is free to use and therefore easily available for other researchers, the researchers have existing knowledge and experience using the platform, and it has a long history and reputation in the field of robotics.

- **Choosing a physics engine.** Open dynamics engine (ODE) is an open source physics engine written as a C library used for simulating rigid body dynamics. ODE was chosen as the appropriate tool to simulate the physics for a few reasons. It is a stable, mature and popular choice for simulating physics in robotic simulations. It includes the ability to simulate multiple joint types, and allows the tuning of friction and contact parameters. Due to its popularity in robotics simulations, Gazebo provides ODE as an option for simulating the physics of the simulation. Since Gazebo handles the majority of the implementation of the library, development is significantly accelerated.

4.1.1 Gripper embodiment in simulation

In order to develop a gripper embodiment in simulation we must consider several other requirements of the gripper, in particular the gripper geometry and the gripper sensing.

Gripper geometry

The simulated gripper developed in this chapter adheres to the description of a suitable gripper given in Chapter 3, i.e. a two-fingered gripper, with the fingers in opposition to each other. Chapter 3 also defines terminology used to describe the parts of the gripper, this labeling convention will be continued in this chapter (see Figure 4.1).

The gripper geometry was realised using an sdf file. An sdf file uses the XML format to describe a collection of links, joints, collision objects, visuals, and plugins which describe the model [208] and can be interpreted by Gazebo. In this way it was possible to define a rectangular link to act as each phalanx of the gripper and join each phalanx with a revolute joint. In addition to the revolute joints, two prismatic joints and six fixed joints were implemented. Gazebo revolute joints allow rotation about one axis, prismatic joints allow translation along one axis and fixed joints allow no relative movement. The first prismatic joint enabled the lateral motion of the gripper which is used by the reactive control strategy and the second to raise the gripper off the ground so as to assess the

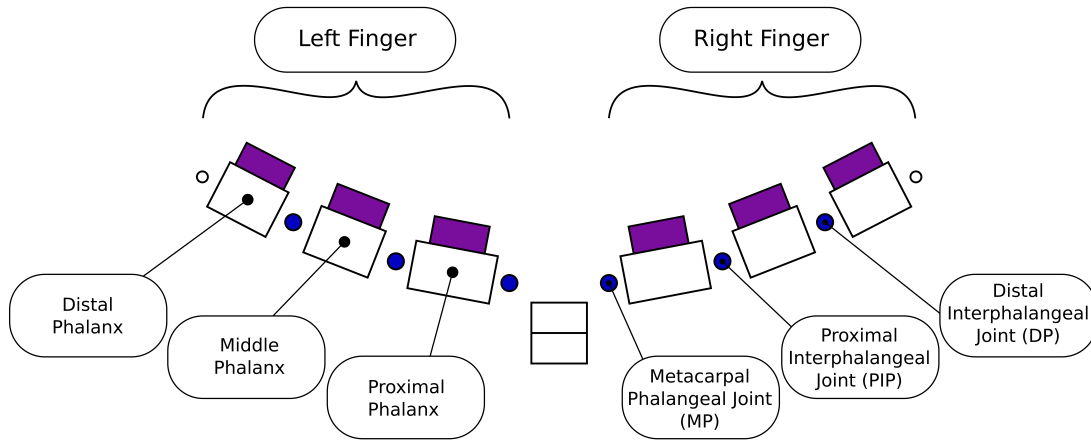


Figure 4.1: Diagram of the simulated gripper, including labels which are used to refer to the different fingers, phalanges and joints

success criteria, i.e. can the attempted grasp support the weight of the ball in the absence of all other external forces. The fixed joints were used to attach the tactile sensors to the gripper, one sensor per phalanx. An illustration of the link tree which is defined in the model file is shown in Figure 4.2. The geometry as defined by the sdf file was based on the measurements of an average male human finger. The critical dimensions used are shown in Figure 4.3.

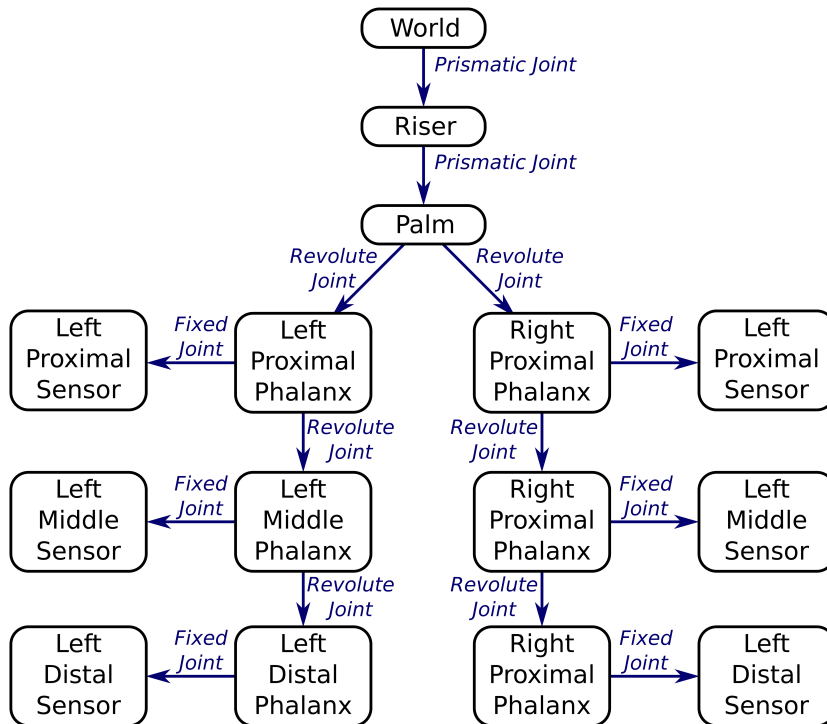


Figure 4.2: Illustration of the link hierarchy described in the sdf file.

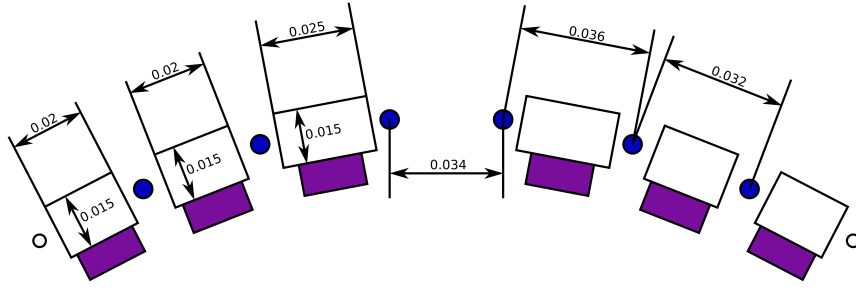


Figure 4.3: Drawing of the simulated gripper showing the critical dimensions. All dimensions are in meters

Gripper tactile sensing

Tactile sensing is critical to the research in this thesis. The design of the sensing system on the embodiment of the simulated gripper has a tactile element on each of the three phalange of both fingers, resulting in six distinct sensing zones spread across the span of the gripper. This deployment of sensors can be seen in the screenshot of the simulated gripper in Figure 4.4.

Tactile sensors were implemented using the built-in '*gazebo_ros_bumper*' plugin, which monitors contacts with each sensor and publishes them in real-time to a ROS topic. The ODE physics engine simplifies contacts between links to point contacts, for this reason each simulated tactile sensor was modeled using multiple discrete Gazebo links. This resulted in multiple contact points per sensor and more stable and realistic performance. Tuned dynamic stiffness (k_p) and dynamic damping (k_d) collision parameters, approximated the deformable nature of silicone-based tactile sensors. Furthermore, non-zero patch and surface radius parameters are used for more accurate simulated frictional behaviour between each sensor and the object.

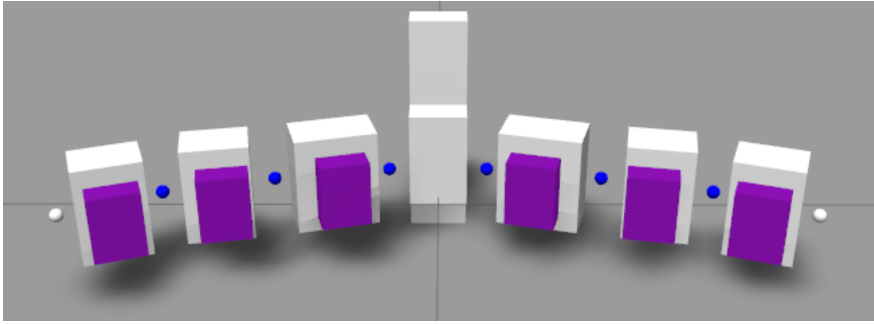


Figure 4.4: Screenshot of the simulated gripper

4.1.2 Simulation of the underactuated mechanism

The actuation systems on the embodiment of the gripper must be suitable to actuate two fingers, each with three degrees of freedom (DoF). It should be highly responsive to react fast enough to grasp the moving object, and it must be able to securely grasp a sphere. A suitable underactuated mechanism was identified as a way to deliver on these requirements.

A system is said to be underactuated if the number of dimensions in which it is free to move, i.e. its DoF, is more than the number of independent actuators. The mechanism chosen couples the rotation of the three DoF on each finger to a single actuator by means of a tendon which is routed through the finger, and applies a moment about each joint. This mechanism has been used on existing grippers developed by the Advanced Robotics Technology and Systems Laboratory (ARTS Lab) [209, 210, 211]. A huge advantage of this method is the way in which it causes the gripper joints to close sequentially. When actuated, the finger starts by moving the MP joint, when this meets an obstacle the PIP joint begins to move, finally when the middle phalanx meets an obstacle the DP joint begins to move. This motion is shown in Figure 4.5 where the gripper is illustrated grasping a ball. This technique is inherently adaptive, the object's position, size and shape affect how the gripper moves, and the design of the mechanism is such that the gripper tends to wrap around the test object. This is illustrated in Figure 4.6 where the grasp shown is asymmetric because it adapted to the off-center position of the object.

The ODE physics engine exclusively simulates rigid bodies, therefore there is no easy way

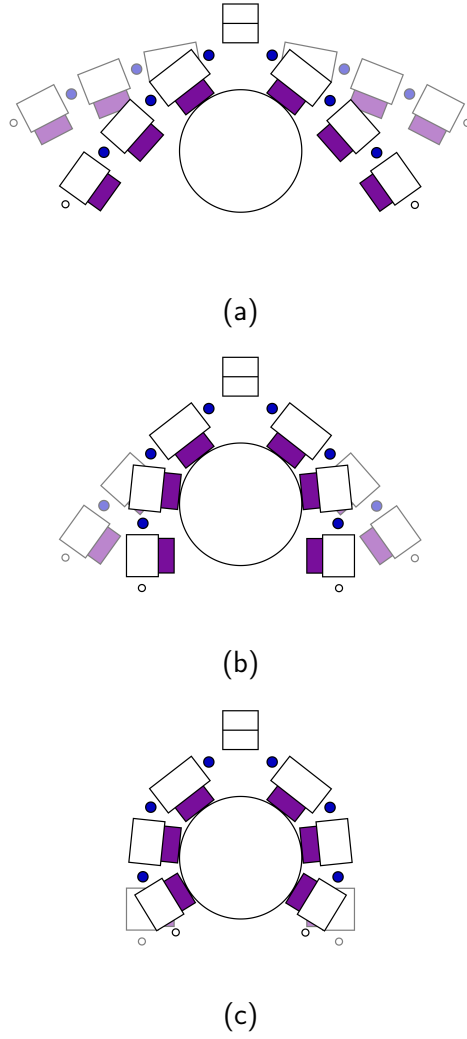


Figure 4.5: Illustration of the sequential closing motion which results from the use of the underactuated mechanism

to simulate a long flexible tendon, like that used when deploying this mechanism on a real-world gripper. To overcome this, a custom Gazebo plugin is developed to emulate the behaviour of the under-actuated mechanism. Each joint is controlled using a velocity, proportional-integral (PI) controller and the torque at each joint is monitored. Initially all joint velocities and target joint velocities are set to 0. When grasping, the plugin provides a step increase in the target velocity of the controller for the first joint, the MP joint. When the torque of this joint exceeds an empirically determined threshold, corresponding to when it meets an obstacle in simulation, the torque on the MP joint remains constant and the controller for the next joint, the PIP joint, is given a non-zero target velocity and begins to move. This is repeated for the PIP and DIP joints. Pseudocode outlining this is

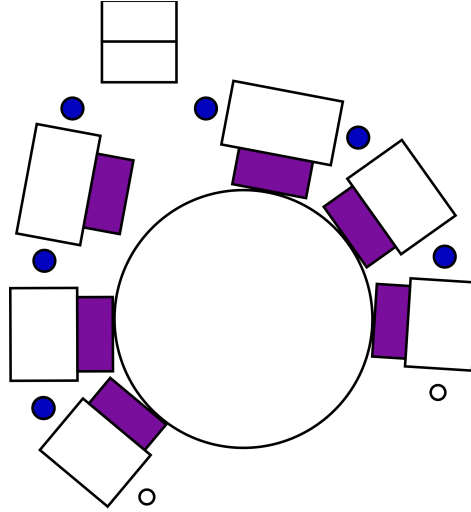


Figure 4.6: Illustration of the final position of the gripper after grasping a ball. The grasp is asymmetrical since the position of the ball is offset from the center of the gripper and the underactuated mechanism adapts the final grasp configuration accordingly

shown in Algorithm 1. This is a direct parallel of how the tendons leverage decreasing mechanical advantage for subsequent joints on real world hardware, and results in a sequential closing motion, and inherently adaptive grasp, characteristic of this type of under-actuated gripper.

Algorithm 1: Pseudocode describing the implementation of the gazebo plugin which simulates the under-actuated closing mechanism.

```

def CloseFinger (finger):
    current_joint = "Metacarpal Phalangeal Joint";
    SetTargetVelocity(current_joint, targetvelocity);
    repeat
        repeat
            torque ← ReadJointTorque(currentjoint)
        until torque > threshold_torque;
        current_joint = MoveToNextJoint(currentjoint);
        SetTargetVelocity(currentjoint, targetvelocity);
    until current_joint == final_joint;
end

```

4.1.3 Control strategies and implementation

In order to deploy the predictive and reactive control strategies on the simulated gripper and conduct performance tests, a bespoke ROS control infrastructure is developed. This ROS control infrastructure is developed as a collection of ROS nodes, which can be

arranged into three categories based on functionality. Actuation nodes interface with Gazebo to control the simulated gripper, monitoring nodes monitor the simulation in order to implement and automate the testing procedure and control strategy nodes implement each of the control strategies so that their performance can be evaluated.

Actuation nodes, are used to interface with Gazebo and control the gripper. They enable mechanisms to control each of the simulated gripper's four virtual actuators. One actuator per finger, the third to control the lateral motion of the gripper and the fourth to raise the gripper off the ground plane and determine if the grasp attempt was successful. The motion of each finger is defined by the underactuated mechanism and managed by the Gazebo plugin outlined in Section 4.1.2. This plugin also advertises a ROS service, which is accessible to other nodes. This implementation allows any ROS node in the network to call the service and command the left, right or both fingers to close. The actuation nodes which control the lateral motion of the gripper and the grasp evaluation procedure are implemented using position controllers. Each node advertises a ROS service, accessible to all other ROS nodes, which commands the actuator. In the case of the lateral motion actuator, the service call requires '*step size*', '*limit*', and '*direction*' parameters to be specified. When commanded, these nodes use the built-in '*/gazebo/set_link_state/*' service to control the position of the joint according to the provided parameters.

Monitoring nodes continuously monitor parameters relating to the current state of the simulation and are used for several reasons. The first use of a monitoring node is related to simulator efficiency. The mechanism provided by Gazebo to monitor the state of the simulation is to request the information via a ROS service call. If multiple nodes require the same information about the state of the simulation, making multiple service requests can cause the simulation to slow down. An example of this is the node which monitors the current state of the ball. These nodes then broadcast this data via a ROS topic, making them available to all other nodes without a service call. The second use of a monitoring node is for diagnostics and data collection. For example, a node was implemented which monitored and recorded the joint positions during a grasp. These joint positions are not part of any of the tested control strategies but are useful for illustrating

and diagnosing a grasp (see Figure 4.7). The final monitoring node is the supervisor node, which was developed to oversee and automate the testing procedure. This node spawns the ball in the appropriate location, sets the ball in motion toward the gripper at the test speed, observes the grasp attempt and commands a grasp evaluation according to the success criteria, finally recording the result and re-spawning the ball for the next test. This node also varied the independent variables from test to test, ensuring that the grasping performance was evaluated under a range of object speeds, object-gripper contact positions and grasp timings.

A control strategy node is developed for each of the strategies under evaluation. The logic associated with each of these is outlined in Chapter 3 Section 3.4, (see Figures 3.5 and 3.6). These nodes used a combination of ROS topics and services, provided by the nodes previously discussed, first to collection information about the current position of the ball, and real-time tactile sensor data, then to determine a gripper response according to the grasping strategy, and finally to actuate the gripper appropriately.

4.1.4 Controlling independent variables

The three parameters previously identified to represent the grasping conditions were: a positional offset from optimum in the initial point of gripper-object contact, a temporal offset from optimum in the timing of grasp initiation, and the object speed.

First, the positional offset was varied from test to test by changing the position where the test object is originally spawned. In this way the initial point of contact can be controlled, and the performance under a range of different contact positions evaluated. The second parameter which must be controlled is the grasp timing. In order to test the performance at a range of grasp initiation times, the optimum time to initiate the grasp must first be empirically determined. With a known optimum time, the grasp can be initiated at a range of temporal offsets from the optimum by monitoring the position and speed of the simulated ball during testing. Finally the object speed must be controlled. The object is set in motion by using the built-in service `'/gazebo/apply_body_wrench'`, provided by

Gazebo. The force applied to the object by this service is described by the vector given to the service as an input. Different object speeds can be tested by tuning this force.

4.1.5 Demonstration of simulated sample grasp

When combined, the features discussed in this section provide a realistic simulation of a robotic gripper grasping a moving object. A sample grasp was performed and relevant data was collected during the attempted grasp, including ball and gripper positions as well as the position of each finger joint. This data is shown in Figure 4.7, with several key features labeled.

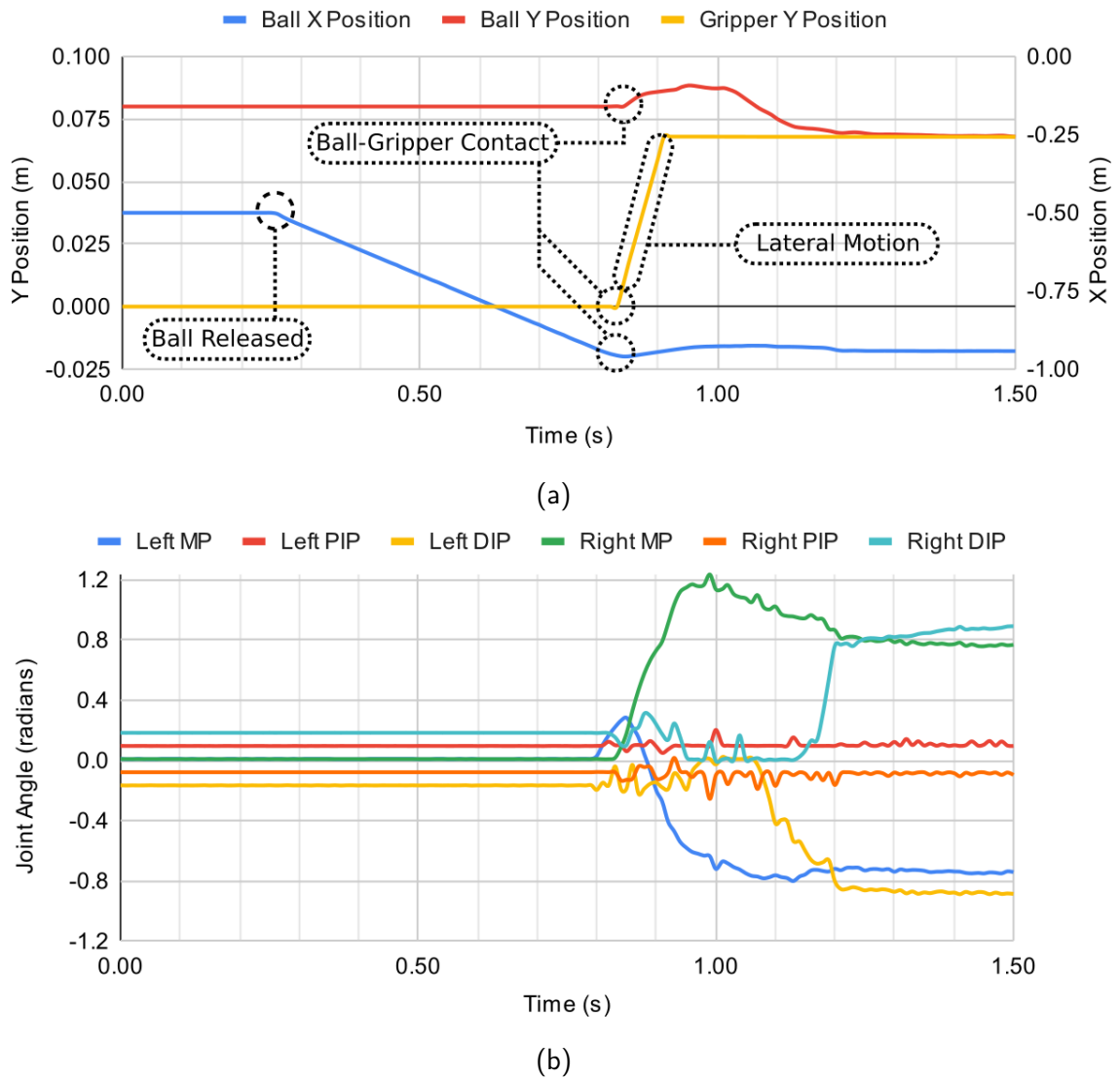


Figure 4.7: Graphs showing data collected during an example of a successful grasp using the simulated gripper. (a) Shows the ball and gripper position and (b) shows the joint angles, during a successful grasp.

4.2 Calibration of the simulation

It is essential that the simulation accurately represents the behaviour of the real-world system it emulates. In order to examine this, a real-world, robotic gripper was developed using the same morphology, sensing systems, and actuation systems as the simulated gripper described in this chapter. This real-world gripper will be discussed further in Chapters 5. Access to a real-world embodiment of a similar gripper enables calibration of the simulated gripper. Controlled grasps were conducted on both virtual and real-world embodiments of the gripper. Both grippers grasped a static object of known dimensions while the joint angles were monitored. Three objects were used, the test object (a sphere of diameter of 57mm) and two cylinders, one smaller than the test object (diameter of 40mm) and one larger (diameter of 70mm). Three grasps per object were conducted and the results averaged. Particular attention was given to two features of the grasp which it was vital to simulate accurately, these were:

- **Gripper kinematics.** The behaviour of this gripper is largely determined by the under-actuated mechanism used. On the real-world gripper this is achieved through a cable driven mechanism, where a single cable is used to actuate multiple joints. Careful design of the mechanical advantage of the cable relative to each joint determines in which order they will move. This causes the finger to close in a characteristic, sequential manner. It is essential that this behaviour is replicated on the real-world gripper.
- **Joint speeds.** Ensuring that the simulated joint speeds reflect those which are appropriate and feasible on a real world gripper is essential in order for the results of the simulation to be applicable to the real world. To ensure this, the joint speeds of the real world gripper were monitored during calibration grasps and matched to the joint speeds during identical grasps in simulation.

Output from the calibration tests are shown in Figure 4.8. It can be seen that the simulation successfully recreated the sequential nature of the joint movement. The mean

velocity of each joint, represented by the slope of the relevant section of the line in Figure 4.8, have been estimated and are shown in Tables 4.1 and 4.2. Table 4.3 shows that the average speed of each joint of the simulated gripper is within one standard deviation of the average speed of their counterpart on the real world gripper (0.49, 0.05, and 0.98 standard deviations for base, middle and tip finger joints respectively).

Table 4.1: Joint Speeds During Real-World, Calibration Grasps

Speed(deg/sec)	Base	Middle	Tip
Small Object	0.620	0.216	0.480
Test Object	0.687	1.114	0.822
Large Object	1.065	0.788	1.214
Average	0.791	0.706	0.838

Table 4.2: Joint Speeds During Simulated Calibration Grasps

Speed(deg/sec)	Base	Middle	Tip
Small Object	0.747	0.494	0.398
Test Object	1.085	0.757	0.809
Large Object	0.499	0.569	0.557
Average	0.777	0.607	0.588

Table 4.3: Difference in joint speeds during real-world and simulated calibration grasps

	Base	Middle	Tip
Δ Speed (deg/sec)	0.108	0.009	-0.481
Δ Speed (σ)	0.49	0.05	0.98

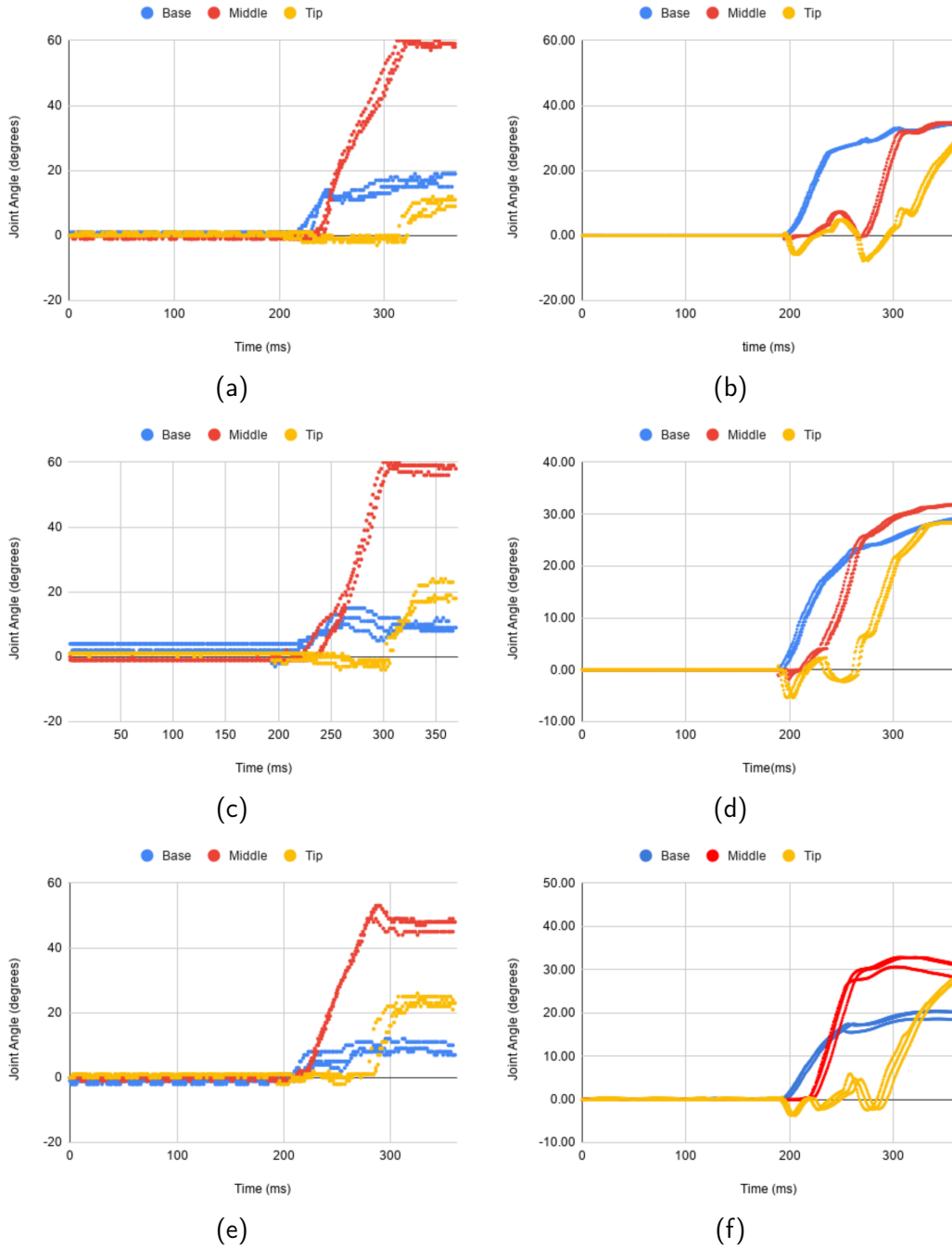
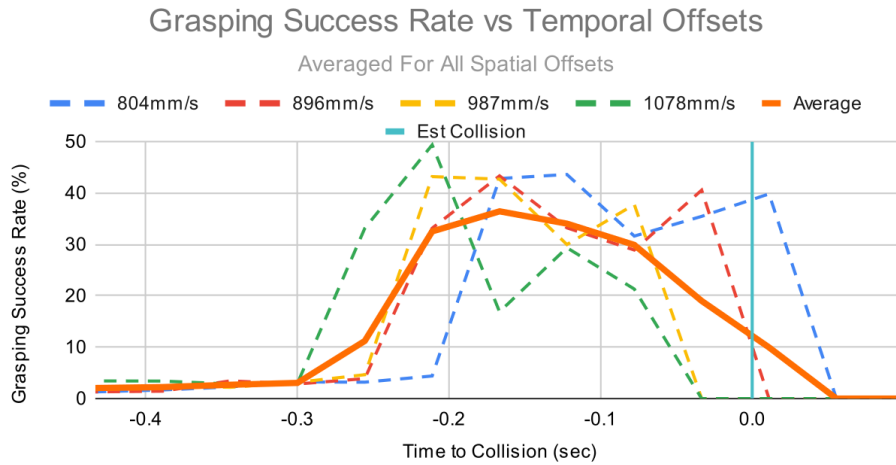


Figure 4.8: Joint angles recorded during calibration grasps (a) Real-world grasp of small object (b) Simulated grasp of small object (c) Real-world grasp of test object (d) Simulated grasp of test object (e) Real-world grasp of large object (f) Simulated grasp of large object

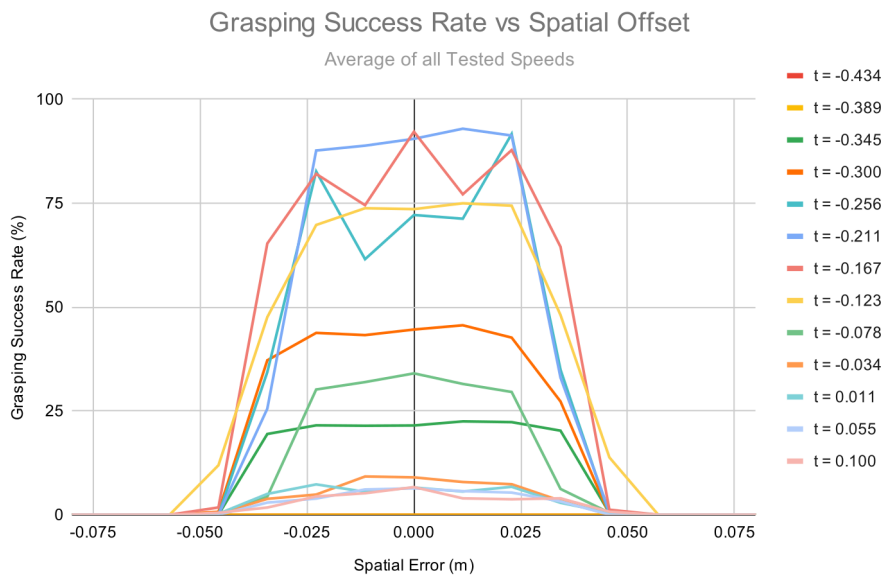
It is also worth noting several differences between the real-world and simulated control grasps. For example, during the simulated test grasps, a small variation in the joint angle can be seen when a joint's parent link is moving but the joint itself has not yet been actuated. This feature does not exist for the real-world system, and is caused by a difference in the dynamic behaviour of each system, and in the mechanism which applies the joint torque. Similarly, the final joint positions in simulation do not correlate to those of the real-world tests. This is caused by an array of minor differences including the initial position of each joint and the material properties of the tactile sensors. These differences are small and do not affect the simulation's ability to accurately represent a grasp.

4.3 Determining the optimum grasp timing

In order to examine the performance of the predictive control strategy under a range of grasp timing errors, it is essential to understand the optimum performance. There are multiple ways in which the optimum time to initiate the grasp can be defined. In this case the optimum grasp timing is defined based on a time before gripper-object contact. To determine this, a series of experiments were performed using the simulated gripper. This involved examining grasp success rates as a function of different ball speeds, ball-gripper contact points and grasp initiation times. The optimal time to initiate the grasp was taken as the point that had the highest overall mean grasp success rate across all grasping conditions (approximately 0.17s before contact). The results from these simulations are given in Figure 4.9.



(a)



(b)

Figure 4.9: Data used to find the optimum time to initiate the grasp, t_{Opt} . (a) Graph showing grasping success rate relative to the grasp initiation time (b) Graph showing grasping success rate relative to the spatial offset for multiple different grasp initiation times

4.4 Experimental Procedure

For each test, the ball was spawned at the desired location, and was set in motion toward the gripper at the test speed. The ball moved toward the gripper and a grasp attempt was made using the relevant grasping strategy. The gripper then rose off the ground plane while maintaining the grasp to determine if the gripper could support the weight of the ball. If it could do so for five seconds the grasp was deemed a success, otherwise it was deemed a failure. This result was recorded and the simulation reset with the new set of experimental conditions.

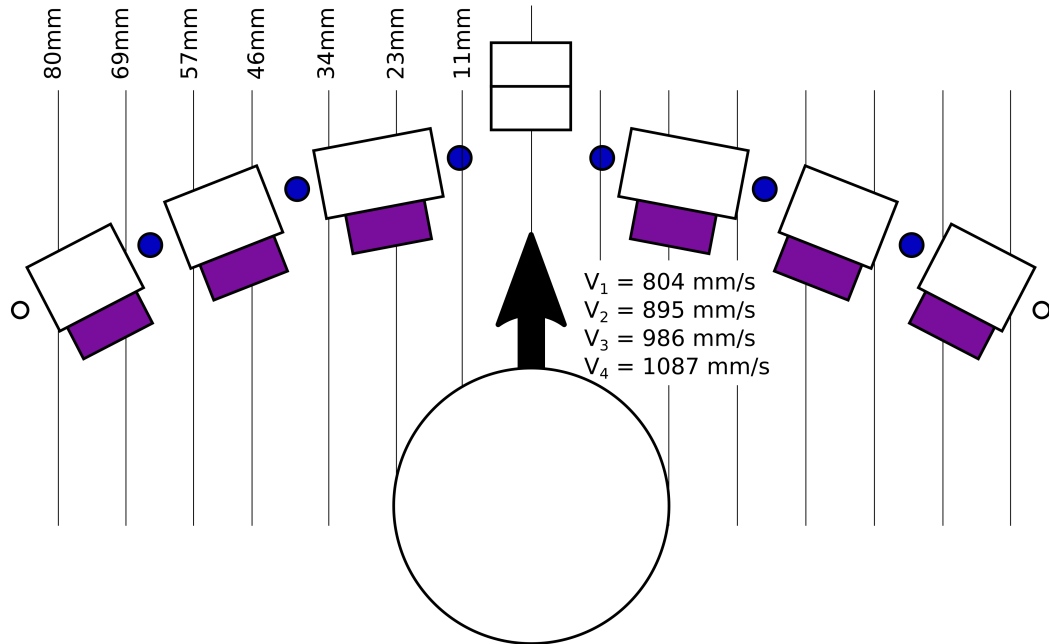


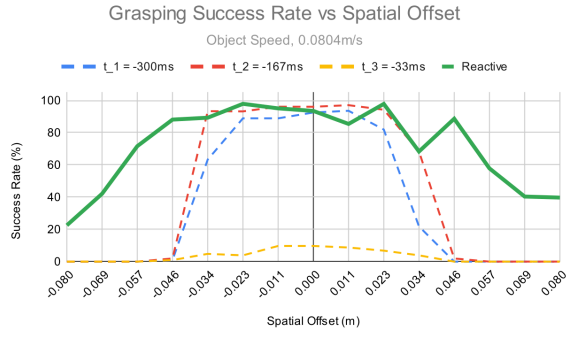
Figure 4.10: Diagram of the simulated gripper including a visualisation of the tested object speeds and object-gripper contact positions.

4.5 Results

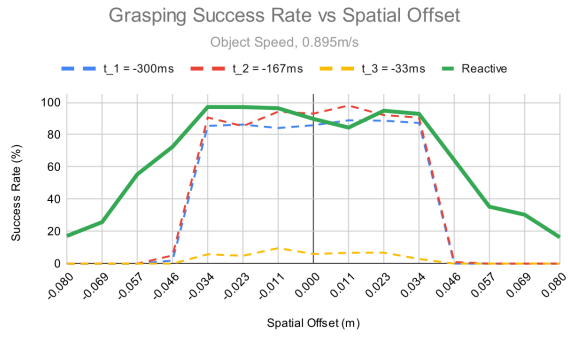
A series of experiments were conducted to investigate the performance of both grasping strategies. Testing was conducted at four ball speeds (0.804m/s, 0.895m/s, 0.987m/s, and 1.087m/s). Fifteen different spatial offsets were tested, equally spaced between -80mm and 80mm, see Figure 4.10. The performance of the predictive control strategy was tested at three grasping initiation times. These were; the time previously determined

to be optimum, a time before the optimum, i.e. a negative temporal error (-0.13 seconds), and a time after the optimum, i.e. a positive temporal error (+0.13). For the reactive control strategy the grasp was initiated once contact with the ball had been detected by one of the gripper's tactile sensors. For each test condition, a minimum of 100 tests were conducted and the results at each ball speed are summarised in Figure 4.11a - Figure 4.11d.

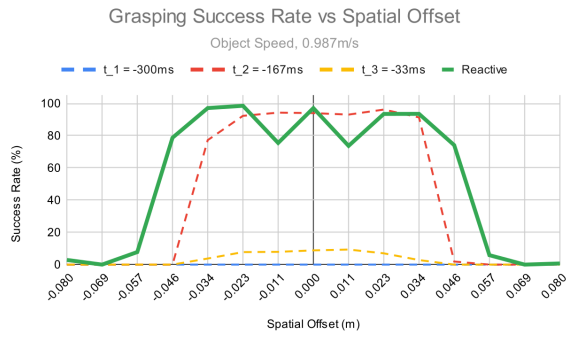
Statistical analysis, using a chi-square test of independence were used to compare the success of the reactive control strategy with each of the three predictive control strategies. These results are presented in Tables 4.4 & 4.5. Due to the symmetry of the gripper, only results from positive offsets (i.e. offsets in the range 0-80mm) are presented. It is observed that the reactive controller matches or outperforms each of the predictive controllers under the majority of conditions. The chi-square test shows this performance improvement to be statistically significant for most of these although not all, this is discussed further in the next section.



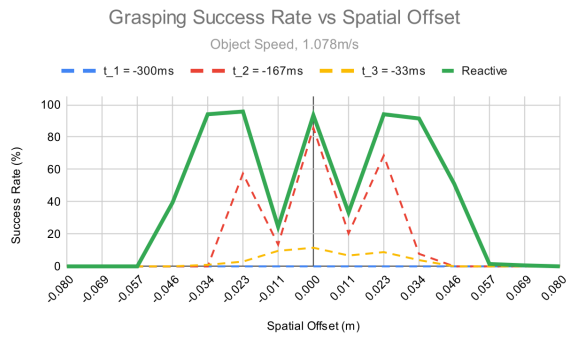
(a)



(b)



(c)



(d)

Figure 4.11: Results for different speeds of the moving object: (a) 804mm/s, (b) 895mm/s, (c) 987mm/s, and (d) 1078mm/s

Table 4.4: Chi-squared analysis of results, part 1 of 2. Where ‘*delay*’ is the grasp time minus the optimum time, ‘*n*’ is the number of tests conducted, χ^2 is the chi-squared value and P_{val} is the probability of obtaining the results presented if the strategy used has no effect on grasping performance. A single asterisk indicates a significant difference in performance at a significance level (α) < 0.05 , a double asterisk indicates a significant difference in performance at $\alpha < 0.01$

Spatial Offset (m)	Speed	Temporal Offset (ms)	Number of Successes		Number of Tests		χ^2	p_value
			Predictive	Reactive	Predictive	Reactive		
0	0.8044	-0.133	99		107		0.0015	0.96911
		0	98	129	102	138	0.3496	0.55434
		0.133	10		103		166.1328	0.00000*
	0.8995	-0.133	91		106		0.5448	0.46045
		0	94	123	101	137	0.4261	0.51391
		0.133	6		101		161.2701	0.00000*
	0.9866	-0.133	0		105		223.3889	0.00000*
		0	94	134	100	138	0.7222	0.39542
		0.133	9		102		186.1479	0.00000*
	1.0777	-0.133	0		107		208.4850	0.00000*
		0	84	130	98	139	3.1600	0.07546
		0.133	12		104		161.2804	0.00000*
0.0114	0.8044	-0.133	102		109		3.4102	0.06479
		0	99	123	102	144	7.9170	0.00490*
		0.133	9		103		138.8327	0.00000*
	0.8995	-0.133	96		108		0.7002	0.40272
		0	101	119	103	141	11.0321	0.00090*
		0.133	7		104		141.4406	0.00000*
	0.9866	-0.133	0		107		134.7183	0.00000*
		0	94	109	101	148	13.7455	0.00021*
		0.133	10		107		100.6731	0.00000*
	1.0777	-0.133	0		107		41.7527	0.00000*
		0	22	46	108	138	4.4625	0.03465*
		0.133	7		104		23.0072	0.00000*
0.023	0.8044	-0.133	86		105		16.6277	0.00005*
		0	98	136	104	139	1.2803	0.25784
		0.133	7		103		199.1194	0.00000*
	0.8995	-0.133	94		106		2.3430	0.12585
		0	94	129	102	136	0.3335	0.56361
		0.133	7		103		181.7474	0.00000*
	0.9866	-0.133	0		107		207.5198	0.00000*
		0	101	129	105	138	0.4134	0.52025
		0.133	7		100		173.5506	0.00000*
	1.0777	-0.133	0		107		209.0371	0.00000*
		0	72	128	105	136	25.6100	0.00000*
		0.133	9		102		170.1109	0.00000*
0.034	0.8044	-0.133	24		110		51.8290	0.00000*
		0	69	97	101	142	0.0000	1.00000
		0.133	4		103		99.6252	0.00000*
	0.8995	-0.133	96		110		1.6642	0.19704
		0	96	131	106	141	0.1867	0.66568
		0.133	3		104		192.1329	0.00000*
	0.9866	-0.133	0		109		210.5185	0.00000*
		0	95	130	104	139	0.1554	0.69343
		0.133	3		106		195.6972	0.00000*
	1.0777	-0.133	0		106		199.3475	0.00000*
		0	8	129	103	141	166.0602	0.00000*
		0.133	4		103		180.7045	0.00000*

Table 4.5: Chi-squared analysis of results, part 2 of 2

Spatial Offset (m)	Speed	Temporal Offset (ms)	Number of Successes		Number of Tests		X ²	p_value
			Predictive	Reactive	Predictive	Reactive		
0.046	0.8044	-0.133	0		107		187.7288	0.00000*
		0	2	125	103	141	175.8517	0.00000*
		0.133	0		103		183.6966	0.00000*
	0.8995	-0.133	0		107		104.7330	0.00000*
		0	1	88	103	137	98.1618	0.00000*
		0.133	0		103		101.7164	0.00000*
	0.9866	-0.133	0		107		133.3696	0.00000*
		0	2	103	104	139	123.3723	0.00000*
		0.133	0		103		129.8664	0.00000*
	1.0777	-0.133	0		106		73.7755	0.00000*
		0	0	71	103	139	72.0126	0.00000*
		0.133	0		105		73.1888	0.00000*
0.057	0.8044	-0.133	0		108		90.1812	0.00000*
		0	0	81	104	140	87.4825	0.00000*
		0.133	0		103		86.8047	0.00000*
	0.8995	-0.133	0		105		44.0892	0.00000*
		0	0	48	103	136	43.3163	0.00000*
		0.133	0		102		42.9293	0.00000*
	0.9866	-0.133	0		106		4.6978	0.03020*
		0	0	8	103	137	4.5417	0.03308*
		0.133	0		101		4.4377	0.03515*
	1.0777	-0.133	0		106		0.2896	0.59048
		0	0	2	100	136	0.2493	0.61757
		0.133	0		101		0.2560	0.61288
0.069	0.8044	-0.133	0		110		56.4552	0.00000*
		0	0	67	105	166	54.1505	0.00000*
		0.133	0		100		51.8298	0.00000*
	0.8995	-0.133	0		109		38.5606	0.00000*
		0	0	51	103	168	36.5549	0.00000*
		0.133	0		104		36.8900	0.00000*
	0.9866	-0.133	0		108		0.0000	1.00000
		0	0	0	103	138	0.0000	1.00000
		0.133	0		103		0.0000	1.00000
	1.0777	-0.133	0		108		0.0000	1.00000
		0	0	1	103	164	0.0000	1.00000
		0.133	0		101		0.0000	1.00000
0.08	0.8044	-0.133	0		108		53.0845	0.00000*
		0	0	56	103	141	50.8709	0.00000*
		0.133	0		106		52.2007	0.00000*
	0.8995	-0.133	0		105		17.0186	0.00004*
		0	0	23	103	141	16.6886	0.00004*
		0.133	0		101		16.3583	0.00005*
	0.9866	-0.133	0		107		0.0000	1.00000
		0	0	1	101	139	0.0000	1.00000
		0.133	0		104		0.0000	1.00000
	1.0777	-0.133	0		106		0.0000	1.00000
		0	0	0	106	138	0.0000	1.00000
		0.133	0		107		0.0000	1.00000

4.6 Discussion

Results show that the predictive strategy performed well at low offsets, both spatial and temporal, but the grasping success rate dropped as the offset increased. This observation reinforces what was forecast, namely high performance when grasping dynamic objects is only possible using a predictive approach when an accurate interception of the object can be achieved. A robotic platform may struggle to achieve the required accuracy when deployed in an unstructured environment, or if its sensing, actuation or computational systems are limited. This effect is amplified at higher object speeds, where sensitivity to errors is larger and grasp performance observed to be worse, even at small induced errors.

In contrast, the reactive grasping strategy demonstrates a robustness to these induced errors, showing the ability to grasp at higher spatial offsets, while matching the performance of the estimate-only strategy at lower offsets. This suggests that the techniques implemented as part of the reactive grasping strategy, i.e. the reactive lateral movement and finger coordination, are able to mitigate the effects of an error in the interception, and increase grasp robustness. This sentiment is reflected in the results of the chi-squared analysis where the performance improvement enabled by the reactive grasping strategy was found to be statistically significant in 65 out of 96 (68%) different combinations of grasping conditions. This became 48 out of 64 (75%) when only considering grasping conditions which have induced temporal errors, and 59 out of 84 (70%) when only considering grasping conditions which have induced spatial errors. These techniques are only possible due to the low bandwidth, low latency, and local nature of the implemented tactile sensing and reactive grasping strategy. This reduces the accuracy required to achieve a successful grasp and is a step toward making such a system effective on a wider range of robotic platforms, deployed in a wider range of environments.

The range of spatial offsets which were tested represents the range at which a successful grasp is possible with this gripper, beyond 80mm the grasping success rate drops to zero, regardless of grasping strategy. To understand the conditions which may lead to

interception offsets in the range tested, an example of a thrown ball was considered. A theoretical analysis of the trajectory was conducted using a purely gravity-based model, neglecting air resistance. It was found that for a ball thrown at a 45° angle toward the robot from two meters away, a 2% error in the measurement of the object velocity, both in magnitude and angle of the velocity vector, when extrapolated will lead to an error in the estimated point of interception of approximately 76mm. This represents grasping under poor conditions where the robot must extrapolate from sensor data collected when the ball is relatively far from the robot. That said, due to occlusion and poor lighting conditions commonly experienced in unstructured environments it is not an unrealistic scenario. Furthermore, this only accounts for errors in the estimated interception point and not errors in the robot's ability to move the gripper to this position.

This research represents the first systematic examination of the effect of different grasping conditions on grasping success rate. Contextualising these results with respect to existing literature is difficult for several reasons. Firstly, the success rates cited in the literature are highly dependant on the robotic system used. In some cases, researchers simplify the problem by removing the robotic gripper entirely and replace it with a cup or net end-effector. The result of this is that the acceptable spatial error tends to be the size of the end-effector used, which has a huge effect on the stated success rate [18].

Secondly, the application driven nature of existing research effects how it is documented, often with an emphases on implementation details and a lack of focus on systematically examining the performance. For example, it is common to cite grasping success rates, without examining how the success rate varies relative to relevant parameters [7, 14]. There are limited examples of robots grasping moving objects with a traditional robotic gripper which cite the interception precision necessary to achieve a successful grasp. One example is research conducted on the Rollin Justin platform, which used a predictive grasping strategy similar to the one implemented in this research. The required interception precision to grasp a ball which is thrown toward it, is stated to be '<2cm' and '<5ms' [6]. The grasping success rate achieved by the Rollin Justin platform at these threshold values are not stated, though it is unlikely an absolute threshold, i.e, 100 %

grasping success rate for all grasps with this or better interception accuracy and 0% in all other cases.

Subsequently, it is difficult to directly compare these results with examples from the literature. In order to enable a comparison between this research and that conducted on the Rollin Justin platform an threshold spatial precision, equivalent to the '2cm' for the Rollin Justin platform, is identified for the system used in this research. The results of testing using a predictive grasping strategy, presented in Figure 4.11, tend to indicate a threshold spatial offset for each speed, above which the grasping success rate drops dramatically, typically resulting in a grasping success rate of less than 10%. This discussion will consider this value to be the required spatial precision for this system, equivalent to the 2cm value stated for the Rollin Justin system.

The results of testing show this significant drop in the grasping success rate for the predictive strategy occurs when the spatial offset is $> 0.034m$ for object speeds between $0.804m/s$ and $0.987m/s$ and when the spatial offset is $> 0.023m$ for the highest object speed of $1.078 m/s$. In comparison the reactive grasping strategy achieves an average grasping success rates of between 72% and 94% under the same grasping conditions. Furthermore, the results of testing on the reactive grasping strategy show that the grasping success rates only experience this same dramatic drop off in performance at an offset of $0.057m$ for the top two speeds, a reduction in the required spatial grasp precision of $0.023m$ and $0.034m$ for speeds of $0.987m/s$ and $1.078m/s$ respectively. Furthermore, the drop in performance with spatial offset is much more gradual for the two slowest object speeds, instead only dropping to between 17% and 22% at the maximum spatial offset tested ($0.08m$). Results of testing show that on average the reduction in the the required spatial precision of the interception is approximately 122%, see Table 4.6. Though it is difficult to directly apply these findings to examples from literature, a similar reduction in the required precision for the Rollin Justin platform would increase the acceptable spatial error from $20mm$ to $44mm$.

The optimum time to initiate the grasp was defined as the time which achieved the

Table 4.6: An analysis of the effect of a reactive grasping strategy on the required spatial precision to achieve a successful grasp. The required spatial precision is defined as the largest spatial offset which results in a grasping success rate greater than 10%

Object Speed (m/s)	Required Spatial Precision		Reduction in required precision	
	Predictive Strategy	Reactive Strategy	(m)	(%)
0.8044	0.034	0.08	0.046	135%
0.8955	0.034	0.08	0.046	135%
0.9866	0.034	0.057	0.023	68%
1.0777	0.023	0.057	0.034	148%
Average			0.03725	122%

highest average grasping success rate, and was determined empirically to be 0.17s before contact with the gripper. However, results suggest that there may be alternative ways in which to define this. Figure 4.9a demonstrates that grasping performance deteriorates quicker when the grasp is early compared to when the grasp is late. An alternative way to define the optimum time would be to select a time from which both positive and negative offsets would result in similar performance. Figure 4.9a also shows that each speed has a distinct optimum time to initiate the grasp. This suggests that the object speed should be included when calculating the optimum time to initiate the grasp. The data collected shows that this approach could increase the average grasping success rate, from 42% when the optimum time is defined as an average of all speeds, to 52% when each tested speed has a distinct optimum time.

In the Figures 4.11c and 4.11d, there is a distinctive drop in performance for low spatial offset, high object speed, test conditions. The grasping motion, as determined by the reactive grasping strategy, was to move the gripper laterally so as to reduce the distance between the center of the gripper and the initial contact point. However, at these low offsets it was found that the rebound resulted in the object crossing the center line of the gripper. This resulted in the lateral movement, triggered by the reactive strategy, taking the gripper further away from the target object, the opposite of the intended effect. This suggests that, despite performance improvements at higher offsets, the reactive strategy presented does not necessarily react optimally and would achieve yet better results through further refinement. One example of how this might be achieved is by increasing

the density of tactile sensing. Higher resolution information about the contact point, would allow the strategy to identify the scenarios in which the object is likely to rebound across the center line of the gripper and trigger a more appropriate response.

An improvement in grasp robustness is shown using a relatively simple tactile sensing implementation. Improvements in both sensor density and the reactive motion have the potential to further improve the gripper performance while grasping moving objects and should be a topic of future research. Further optimization of the time to initiate the grasp, in particular a strategy which considers the effect of the object speed, should also be explored in future research. The simulated environment presented in this work provides an ideal test-bed for tackling these questions. Current modeling is conducted in simulation and while this is valuable for initial testing and development of the hypothesis, validation of the results presented here in a real-world experiment is essential and is addressed in Chapter 5.

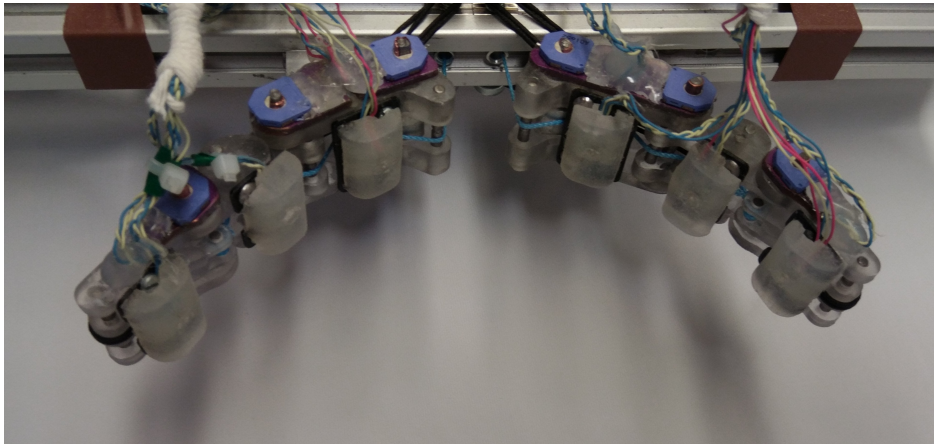
5 | Validation through Real World Testing

Previous chapters have: developed an experimental methodology to quantify the performance of a gripper grasping a moving object, proposed a heuristic-based, reactive control strategy, and demonstrated the improved robustness offered by this reactive strategy through testing in simulation. This chapter aims to develop a real-world testing apparatus, to deploy the predictive and reactive grasping strategies on a real-world robotic gripper, and to verify, in the real world, the results obtained from simulated testing. To achieve this, a testing apparatus, capable of implementing the experimental methodology outlined in Chapter 3, was developed. This include a two-finger gripper and mechanisms to systematically vary key parameters of the grasping conditions; the position of initial gripper-object contact, the grasp initiation time, and the object speed. Predictive and reactive gripper controllers were deployed on a microcontroller which controlled the gripper. Finally, testing which evaluated the performance of the real-world gripper under a range of grasping conditions was conducted.

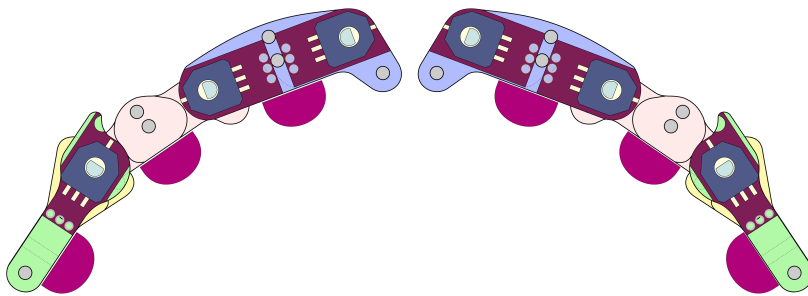
5.1 Development of a real-world gripper

While innumerable gripper design and associated implementations have been published in the prior literature, this research has its own unique set of requirements, for which a custom robotic gripper was developed. A custom gripper design ensured not only that the gripper was able to address each of the requirements, outlined below, but maximised the

usefulness of the gripper by ensuring it could be repaired, its design could be modified, its sensing modalities updated, its actuation systems changed, etc. The same level of adaptability can be difficult to achieve using an off-the-shelf gripper. The gripper shown in Figure 5.1 was designed and manufactured. This process is outlined below.



(a)



(b)

Figure 5.1: Bespoke gripper developed for real-world testing, represented as (a) a photo of the gripper and (b) a drawing of the gripper

5.1.1 Requirements of the real-world gripper

Informed by prior designs, a list of key criteria of the gripper was formulated. These consider, not only suitable for testing, but aim to maximise its usefulness to the project as a whole.

- **Gripper geometry.** The experimental methodology, outlined in the Chapter 3, calls for a two-finger gripper, with fingers in opposition to each other, which can

execute a power grasp to grasp a moving ball. Furthermore, each finger should replicate the kinematics (3 DoF) and dimensions of a human finger.

- **Gripper actuation systems.** The gripper's ability to grasp a moving object should be dependant upon the independent variables like control strategy, and grasping conditions, and not limited by the hardware itself. Low latency and high joint speeds are required to achieve this.
- **Gripper sensing systems.** The gripper should have tactile sensing embedded in the fingers and joint position sensing. Tactile sensing is required to enable the adaptive grasping motion in the reactive control strategy. Joint position sensing, while not used in either of the strategies to be tested, is essential for diagnostics, grasp visualisation and test initialisation.
- **Gripper computation systems.** The conclusions drawn from this research must be applicable to a wide range of robotic platforms, including computationally limited robotic platforms. The computational requirements of sensor analysis, strategy implementation and gripper control should be conducted by a computational unit on-board the gripper which is realistic for a computationally constrained robotic platform.
- **Control strategy implementation.** The gripper must enable the deployment of the control strategies being assessed.
- **High gripper robustness.** The gripper is used and reused in many different experiments over the course of the project. These tests often require hundreds of testing cycles, to ensure sufficiently large sample sizes. The gripper should therefore be sufficiently robust to endure thousands of collisions with an object without breaking or suffering performance degradation.
- **Gripper modularity and reconfigurability.** Modularity and reconfigurability are design philosophies which are emphasised during the design of the gripper. This allows the gripper to be adapted if necessary and lowers the cost of repairs by

allowing individual parts to be replaced.

- **Gripper cost.** The gripper should be relatively inexpensive to aid reproducible and make research in this area accessible.

The remainder of this section discusses these criteria and how they are addressed.

5.1.2 Real-world gripper geometry

The morphology chosen for the embodiment of the real-world gripper is consistent with that identified in the methodology and simulated in prior testing, see Chapters 3 and 4 respectively. Similarly, the terminology used to refer to the different parts of the gripper is the same as that used in previous chapters and is based on the terminology commonly used to refer to a human finger. This is illustrated in Figure 5.2.

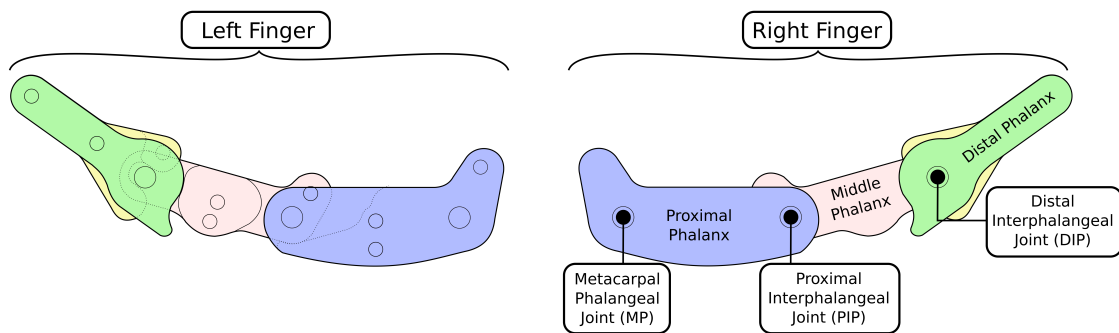
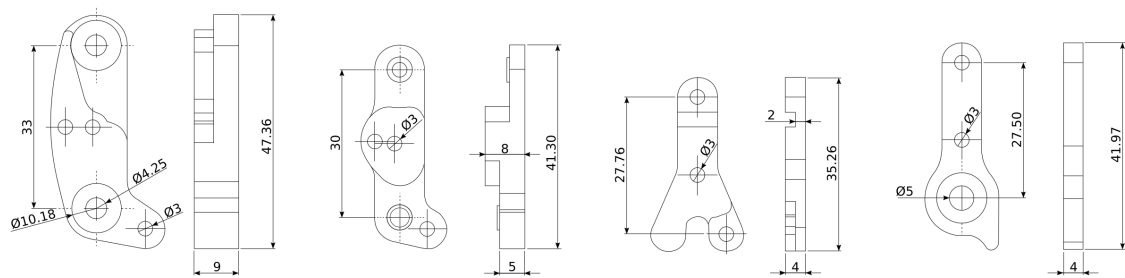
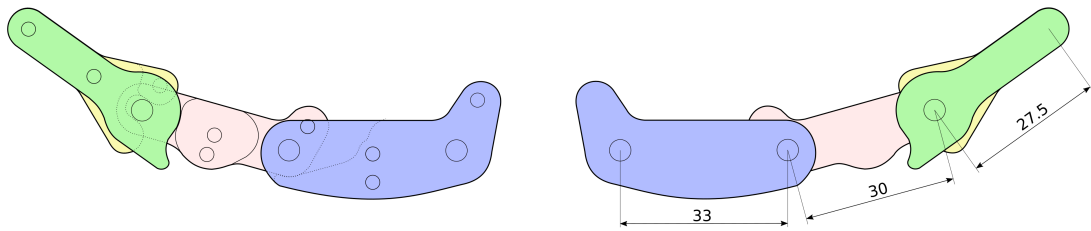


Figure 5.2: Terminology used to refer to parts of the gripper

The critical dimensions of the four distinct parts used to create the gripper as well as the dimensions of the final assembled gripper are shown in Figure 5.3



(a)



(b)

Figure 5.3: Drawings of the final gripper design, showing key dimensions of **(a)** the 4 unique parts used to make up the gripper, and **(b)** the assembled gripper. All dimensions are in millimeters.

5.1.3 Design of the actuation systems for the real-world gripper

To achieve the responsiveness and joint speed requirements, low friction bearings are used on each joint. Furthermore, mass reduction is prioritised during design. The skeletal design, and material choice both contribute to this goal. The actuators used are DC servo motors which are not integrated into the fingers but instead actuate the fingers via a tendon, maintaining the light weight finger design while not compromising actuator velocity and torque. This is a common technique in gripper design [143].

The actuation systems on this gripper can be categorised into two distinct groups, finger actuation and lateral movement. These are discussed in more detail below.

Finger actuation

Underactuation was determined to be beneficial in this application. The underactuation mechanism used allows the three joints of each finger to be moved by a single actuator, a DC servo motor, and is similar to the mechanism used by the Advanced Robotics Technology and Systems Laboratory (ARTS) Lab in several of their robotic grippers [209, 210] including iCub [211], though without the use of pulleys.

A tendon is routed along the finger in a way which applies a moment about the axis of each joint when the tendon is tensioned, as shown in Fig 5.4b. The net moment, M^{net} about each joint is the anti-clockwise moment, M^{ACW} , applied by the tendon, minus the clockwise moment, M^{CW} , applied by the elastic element which couples each link to its parent link and the force applied by any object which the link comes into contact with. Equations 5.1 to 5.4 show how to calculate the net moment about each joint (i).

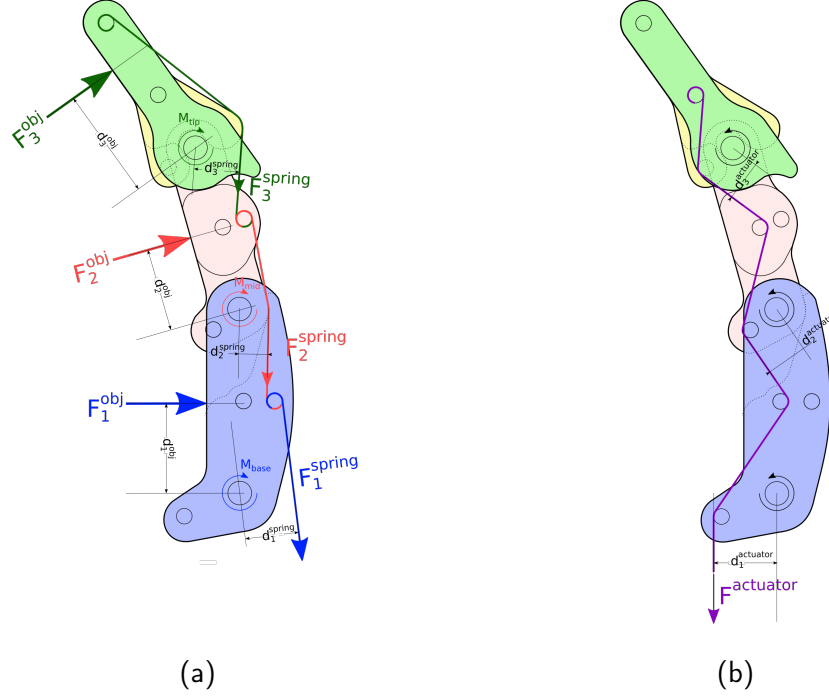


Figure 5.4: Breakdown of the moments experienced by the gripper during grasping, categorised by **(a)** clockwise moments and **(b)** anticlockwise moments

$$M_i^{net} = M_i^{CW} - M_i^{ACW} \quad (5.1)$$

$$M_i^{CW} = F^{actuator} \times d_i^{actuator} \quad (5.2)$$

$$M_i^{ACW} = (F_i^{spring} \times d_i^{spring}) + (F_i^{obj} \times d_i^{obj}) \quad (5.3)$$

$$M_i^{net} = F^{actuator} \times d_i^{actuator} - ((F_i^{spring} \times d_i^{spring}) + (F_i^{obj} \times d_i^{obj})) \quad (5.4)$$

The sequence in which the joints will close can be controlled through design of how the tendon is routed, i.e. the values of $d_1^{actuator}$, $d_2^{actuator}$, and $d_3^{actuator}$ and how the joint springs are implemented, i.e. the values of F_1^{spring} , F_2^{spring} , and F_3^{spring} . Each joint starts to move once the clockwise moment from the tendon exceeds the anticlockwise moment from the spring for that joint, i.e. $M_i^{net} > 0$. The gripper design is such that the tendon tension required to move the MP joint is smaller than that required to move the PIP joint which is similarly smaller than the tension required to move the DIP joint. During a grasp, the tendon tension will continuously increase as the actuator pulls on the tendon. At some

threshold ($M_1^{net} > 0$) the MP joint will start to move. The tendon tension will experience a step increase when the proximal phalanx meets an object, or is otherwise prevented from rotating. This causes the tension to rise above the threshold of the PIP joint, which will subsequently begin to move. Similarly the tendon tension required to move the DIP joint will be exceeded when the middle phalanx meets an object. This sequential joint movement, visualised in Fig 5.5, enables the gripper to self-adapt to object geometry without a complex control strategy [212]. There are also forces caused by joint limits which are not considered during this analysis, these drop to zero during the grasping motion and therefore do not effect the part of the grasp which is considered here.

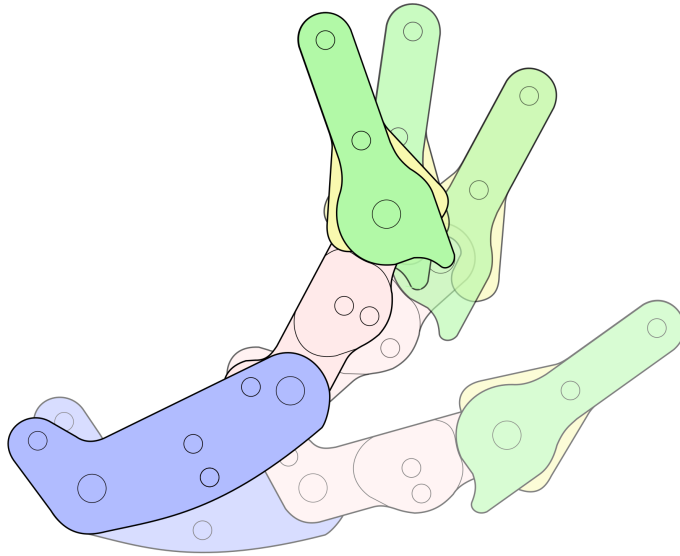


Figure 5.5: Illustration of the gripper at various stages throughout the closing motion, showing the sequential closing motion

Lateral movement

The lateral motion of the gripper is an essential aspect of how it might adapt its grasping motion. On a more complex robotic platform this motion is possible since the gripper is typically deployed at the end of a robotic manipulator, however this is significantly simpler on the 2D testing platform used in this research. For this setup the gripper was attached to the fixed based via a low-friction, linear rail along which the gripper could travel.

Furthermore, this arrangement provides virtually no compliance in the gripper mounting in the direction of ball movement, removing this as a factor in grasping robustness. This additional DoF was actuated using a servo motor, and coupled by a rigid linkage. The lateral movement of the gripper is visualised in Figure 5.6.

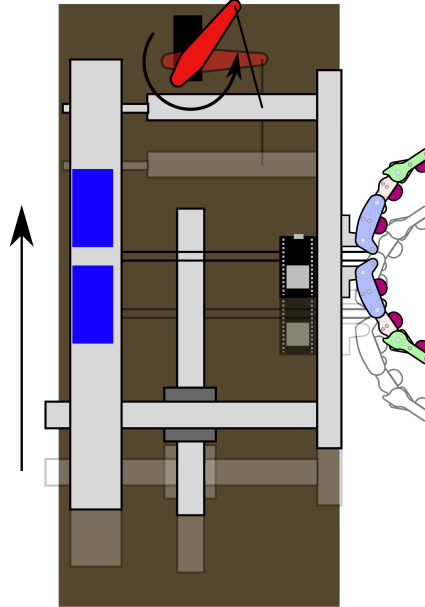


Figure 5.6: Visualisation of the lateral motion of the gripper

5.1.4 Design of the sensing systems for the real-world gripper

There were two sensing modalities implemented on this gripper, joint position sensing and tactile sensing. Joint sensing is implemented using a potentiometer on each of the six joints. Tactile sensing is implemented as six contact sensors, one located on each of the three phalange for both fingers. This is the same implementation as the gripper embodiment in simulation developed in Section 4.1.1.

Joint position sensing

The potentiometer implemented on the gripper was the SV01A103AEA01R00 by Murata Electronics [213], using a modified CJMCU-103 breakout board to simplify the

implementation. An image of the sensor and the implementation of the sensor on the gripper can be seen in Figure 5.7. The thru-hole rotor of the potentiometer is located on the side of the phalange, two on the proximal phalanx and one on the distal phalanx, co-axial with each joint pin. The potentiometer is adhered in place such that the joint pin's orientation and the orientation of the body of the potentiometer are coupled to different phalange. Furthermore, the pin's orientation must be such that the discontinuity in its resistance to angle relationship is outside of the joints range of motion.

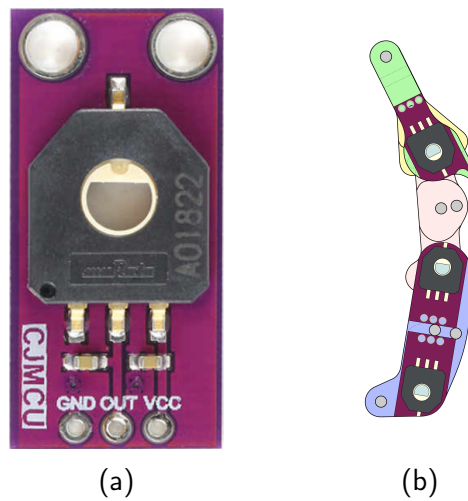


Figure 5.7: Joint sensing used on the bespoke gripper, (a) The potentiometer and (b) the implementation of the potentiometer on gripper

Tactile sensing

The tactile sensing fabricated and used in this research took inspiration from examples seen in the literature [58, 214]. It uses the Hall effect to monitor the position of a magnet relative to a Hall effect sensor. The magnet is embedded in a silicon rubber and placed over a Hall effect sensor, such that contact with the rubber will cause a deformation and a proportional change in magnet-sensor relative position. An air gap between sensor and silicone is used to increase sensitivity, as described by Jamone et al. [215].

The material used was a "Polycraft T-15 Translucent Silicon Rubber" [216]. It is described by the manufacturer as a "two-component, high strength, flexible, moulding compound". It was chosen for this application, since it is sufficiently robust so as to not deteriorate or change in performance after successive collisions during testing. It is

sufficiently flexible, and of a suitable elasticity to act as a medium to suspend the magnet, such that a small force contacting the surface causes a detectable change in the relative sensor-magnet position. Finally, since it was a two-part compound it was easy to mould to a custom shape. The silicone remained liquid for approximately 20 minutes after the two parts were mixed. This was sufficient time to fill and close the mold.

The Hall sensor IC used was a Melexis, MLX90393 Hall effect sensor [124]. This sensor has been a popular choice in literature as it has several characteristics which make it advantageous in this application including:

- **Small.** The MLX90393 is 3x3x1mm. This is particularly important since the tactile sensing is embedded in a robotic finger of comparable size to that of a human finger.
- **Simple.** The sensor itself has built-in logic and can be interfaced with via the I2C communication protocol. Parameters such as the sample frequency, mode (burst, single measurement and wake-on-change), sensitivity, gain, etc. can be set and data collected using an appropriate microcontroller. Furthermore the I2C communication protocol allows two communication lines (SCL and SDA) to enable communication between several devices using 7-bit addressing. This reduces the number of wires which require routing through the gripper.
- **High Sensitivity.** The maximum sensitivity of the sensor (in the Z axis) is $0.294 \frac{\mu T}{LSB}$. This is orders of magnitude more sensitive than necessary for this application. The sensor is tuned, via the on-chip gain and sensitivity, so as the output is within a suitable range for the application.
- **Multi-axis.** The sensor in question reports the magnetic field strength in all three axis (X, Y and Z). This is incredibly useful for tactile sensors implemented in a robotic gripper since it has the ability to report more than just forces normal to the sensor but can also infer information about forces in other directions, i.e. shear. Theoretically this would give a gripper the ability to infer an objects weight, and surface characteristics as well as detect slippage and the angle of incidence of a collision to name a few examples.

- **Sample Frequency.** Since this project is exploring the grasping of moving objects the ability to sample sensors at a sufficiently high frequency is of upmost importance. The sample frequency of this sensor is dependant on a number of variables and settings including the sensor mode, relevant axis, the "BURST_DATA_RATE" parameter, temperature, over-sample-rate of the ADC, etc. For this reason it is difficult to give a value for the max sample rate however it is in the order of tens of milliseconds per sample or 10-100Hz which is sufficient for this application.

One feature of the design of the tactile sensors taken directly from the literature is the use of an air gap between the sensor and magnet-silicone [58, 215]. An observation both from early stage testing of a prototype sensor and from the literature is that there is cross talk between the different axes when the silicone is deformed. Cross talk is when a force normal to the surface (in the z-axis) would cause significant changes in the magnetic field in the x and/or y axis. It was discovered that the reason for this was the silicone which, though elastic, was relatively incompressible. Therefore when a force would deform the compliant covering, the magnet would move both in the direction of the force but also in a direction perpendicular to the force because of the flow of the material. The solution to this was to add an air gap above the sensor. Since air is compressible, and also free to flow out of the cavity, the relative movement between magnet and sensor was more representative of the force causing the deformation. Furthermore the addition of an air gap significantly increased the sensitivity of the sensors. This effect is illustrated in Figure 5.8. A downside of the air gap is the increased hysteresis, however since this research is concerned only with applying a force to the sensor, and not the sensor's behaviour as the force is removed, this is not an issue.

The sensor required two main fabrication methods, the printing of the PCB and the moulding of the silicone rubber. The PCB, seen in Figure 5.9, is a custom PCB which enables the I2C communication protocol and minimises all other wires. The manufacturing of this part was outsourced.

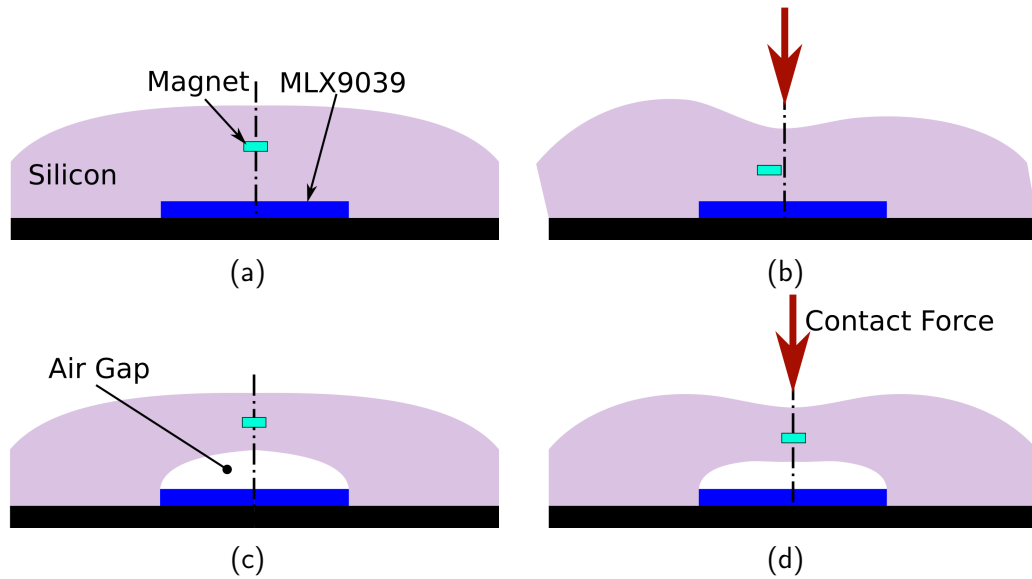


Figure 5.8: A demonstration of the advantage of implementing an air gap in the tactile sensors. This increases sensitivity while reducing cross-talk between different axis. (a) Without an air gap (b) Incomprehensibility causes unpredictable material flow (c) Introduction of an air gap (d) Displacement is in direction of the contact force

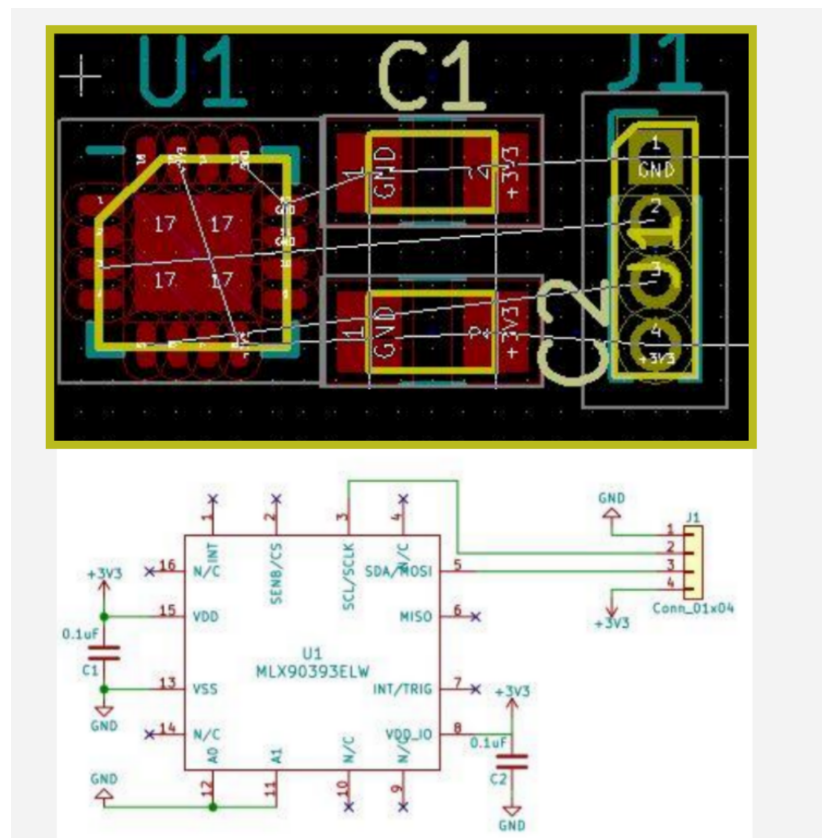


Figure 5.9: The Hall effect sensor used to obtain the tactile feedback was the MLX90393. This shows the schematic of the PCB used.

The silicone was moulded to a custom shape. To manufacture this, molds were created from polylactic acid (PLA) using additive manufacturing. 3D renderings and photographs of this process are shown in the Appendix in Figures A1.1 & A1.2. The mold was made in two parts, the primary mold and a lid. The lid could be bolted into place and included the geometry to create an air gap and auto-alignment with the primary mold. The lid also had risers to allow excess silicone to flow out of the mold, ensuring that there were no unintentional air bubbles left in the silicone during the curing process. In order to hold the magnet in position during the curing process a small (1mm) drill bit was used to create a hole in the bottom of the mold. The magnet could then be attached to the top of the drill bit and the drill bit friction fitted into place.

A drawing and photograph of the real-world gripper include both joint position and tactile sensing can be seen in Figure 5.1

5.1.5 Selection of the computational systems for the real-world gripper

The ESP32 microcontroller, shown in Figure 5.10, was chosen as the sole computational unit which was used to operate the test gripper. The ESP32 provides an inexpensive, low power, small footprint solution which is perfect for robotic applications. Its ability to conduct parallel operations using its on-board co-processor is also very attractive to applications like grasping a moving object, where minimising latency is essential, and there are several independent operations which can be conducted in parallel. Reliable development support, including an active online community, enabled faster development and the wide range of compatible communication protocols, including UART, bluetooth, serial, CAN and WiFi, adds to its suitability as a robotics tool.

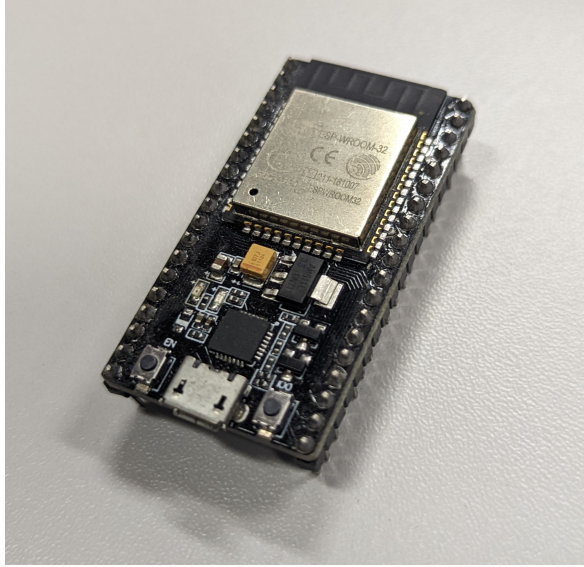


Figure 5.10: The computational unit chosen for the bespoke gripper was the ESP-32 microcontroller, specifically the Node MCU ESP-Wroom-32 Development Module

5.1.6 Control strategies and implementation

Control strategies were implemented on the ESP32 in the form of a precompiled application which was flashed to the microcontroller. This was achieved using the official IOT development framework provided by Espressif, the manufacturers of the ESP32 microcontroller. This development framework is called the ESP-IDF.

In order to leverage the efficiency advantages offered by the microcontroller's coprocessor, the necessary functionality was developed as multiple distinct and independent 'tasks' such that they could be processed by the microcontroller in parallel. A priority was then assigned to each task to enable the microcontroller to determine how to assign computational resources. These tasks are described below.

Supervisor

This task launched when the microcontroller was powered on or reset. It handled the memory resources as well as creating, launching and assigning priority to other tasks. It is responsible for the initialisation of the gripper in preparation for the test to begin and the control strategies being tested. These were the traditional, predictive strategy and the proposed, reactive strategy, and are outlined in Chapter 3.

Read from Tactile Sensor

This task is responsible for interfacing with the tactile sensors and writing the most recent tactile sensor values to a space in the microcontroller's memory. The MLX90393 Hall effect sensor, which is the transducer in this tactile sensor assembly, uses I2C to interface with a controller. This task therefore involved developing a bespoke library to allow the microcontroller to interface with the sensor. This task was only launched during testing of the proposed reactive strategy since the traditional, predictive strategy does not utilise tactile sensing.

Read from Joint Sensor

Responsible for reading the current position of the potentiometer, and therefore the joint, this task was used during the gripper initiation to ensure that the initial joint positions fell within acceptable limits.

Control Finger

There were two instances of this task launched during each test, one to control each finger. This task, when commanded by the supervisor task, would actuate the fingers causing them to open or close.

Monitor Light Gate

This task was responsible for communicating with the light gates. The predictive strategy relied on an estimate of the best time to initiate the grasp, this was provided by a set of light gates. The light gate mechanism used will be discussed further in Section 5.2.4.

Control Lateral Motion

A key adaptive motion used by the reactive control strategy is the lateral motion of the gripper. This task was responsible for receiving speed, limit and direction parameters from

the supervisor node and commanding the actuator to make the corresponding movement.

5.1.7 High gripper robustness

To attain this level of robustness, sheet polycarbonate and a skeletal design is used. Each phalanx was made up of at least two (four in the case of the distal phalanx) polycarbonate components running the length of the phalanx on each side. These were joined using steel pins to create a phalanx and each child phalanx was connected to its parent with a steel pin and two low friction bearings. This design, shown in Figure 5.11, was extremely strong, robust while remaining relatively small in size and light weight.

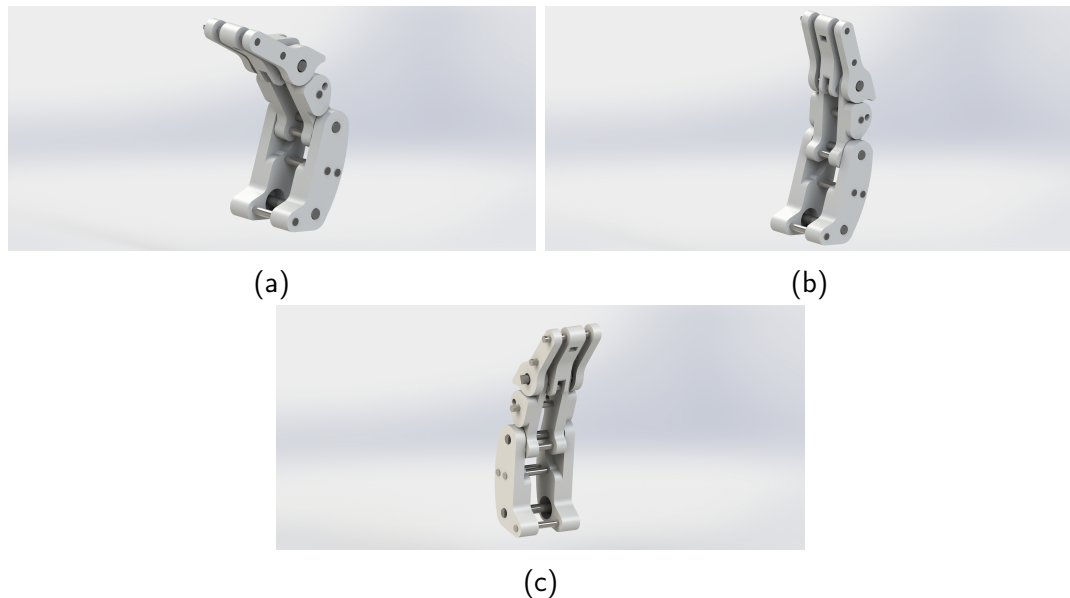


Figure 5.11: Renderings of the finger design. **(a)** Assembled Finger, View 1 **(b)** Assembled Finger, View 2 **(c)** Assembled Finger, View 3

5.1.8 Gripper modularity and reconfigurability

Each finger was made up of eight polycarbonate components of four unique designs, and eleven steel pins of various lengths, see Figure 5.12. To remove, replace or upgrade a part, removing the pins would allow simple disassembly and reassembly. The design of each phalanx also left space between the structural elements. In this iteration of the gripper it is used for routing wires and tendons, however since this space is protected from any

damage that might result from contact between the gripper and an object, it could equally be used for a host of additional sensing and actuation functionality as future applications demand.

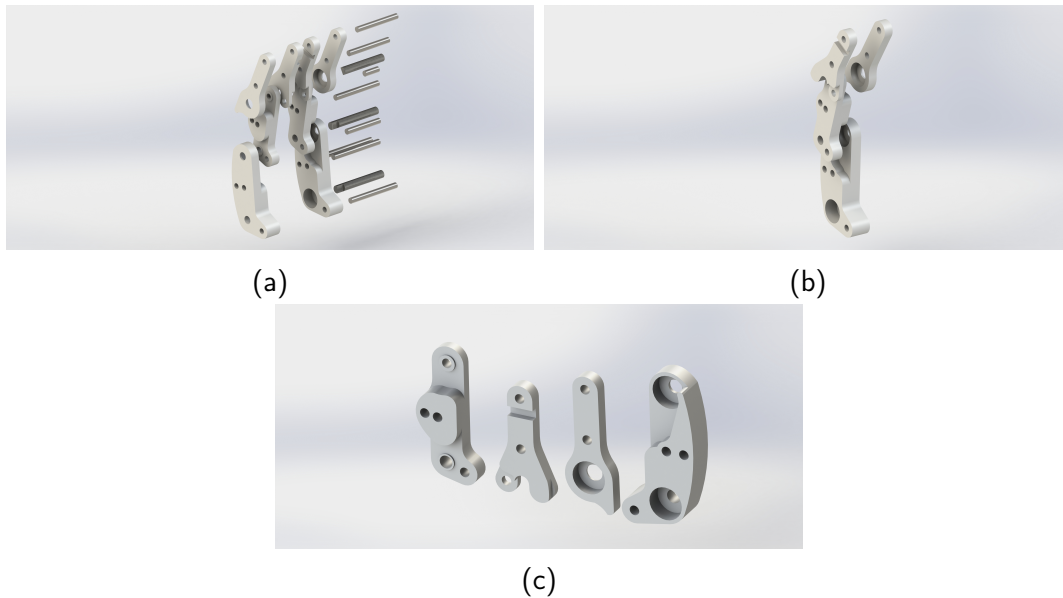


Figure 5.12: Renderings of the finger design. **(a)** Exploded View **(b)** Exploded view of 1 side of the finger **(c)** 4 unique parts required.

5.1.9 Gripper cost

To address the cost requirement the gripper should be manufacturable with readily accessible materials and resources. This ensures that the project does not suffer from any long delays due to the need to replace a damaged part or to modify a design feature. It also lowers the cost of development, enabling several iterations of the gripper to be easily and inexpensively created should the first version fall short of demands. The primary manufacturing methods available and used during this project are additive manufacturing via a Fused Deposition Modelling (FDM) printer and subtractive manufacturing via a 3-axis, CNC milling machine. The entire gripper was designed to be manufactured with sheets of polycarbonate, PLA, steel pins and low friction bearings.

5.2 Development of a real-world testing apparatus

The primary role of the testing apparatus used in this chapter is to enable control over the experiment's independent variables. This involved the development of mechanisms which vary key parameters of the grasping conditions in a highly repeatable manner. These key parameters are: the position of the initial object-gripper contact point, the grasp timing and the object speed.

5.2.1 Choice of test object

Foremost among the design decisions to be taken is the choice of test object. The object chosen for the experiment was a billiards ball (diameter=57mm, mass=166g).

The choice of a spherical object has already been discussed in Chapter 3 and prior testing in simulation, see Chapter 4, also used a spherical object. This specific object was chosen since its mass provided a good test of grasp effectiveness, there was a high level of confidence in the quality of its manufacture, and it could be easily sourced by other researchers.

5.2.2 Controlling object velocity

First, the direction of the object's velocity vector should be considered. An inclined track was used to set the ball in motion and was implemented using a metal bar with a U-shaped cross-section that constrained the motion of the ball to 1 degree of freedom, see Figure 5.13. In this way the velocity vector was highly controlled.

Next, consider the magnitude of the velocity vector. This is dependent on its release height on the inclined track, releasing from a greater height allows more time for acceleration, resulting in a higher speed at the point of contact. At the base of the incline, the ball is transitioned to a horizontal, planar surface. At this point a pair of light gates are used to calculate the test speed according to Equation 5.5.

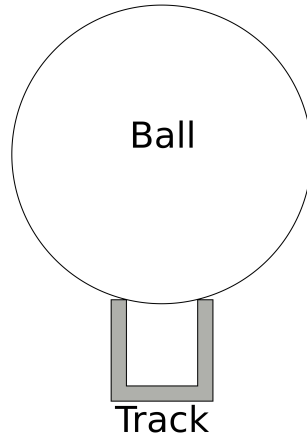


Figure 5.13: Ball trajectory is controlled by an inclined track. A U-channel aluminium extrude (shown in cross-section) is used as the track.

$$ball\ speed = \frac{\Delta x}{\Delta t} \quad (5.5)$$

Where:

Δx is the distance between the light gates

Δt is the time between triggering the first and second light gates

Servo motors provide a highly-repeatable release mechanisms which set the ball in motion.

The repeatability of this approach is validated, see Table 5.1; two hundred tests conducted at three test speeds show a relative standard deviation of just 1.6%, 1.3%, and 2.0% respectively.

(m/s)	Speed 1	Speed 2	Speed 3
Mean	0.82363	0.96363	1.08727
Standard Deviation	0.01305	0.01252	0.02127
Standard Error	0.00092	0.00089	0.00150
Number of Measurements	200	200	200

Table 5.1: Results of experiments to validate the repeatably of controlling and measuring the object speed. 600 tests were conducted at 3 different speeds

5.2.3 Controlling the position of initial gripper-object contact

The mechanism by which the initial gripper-object contact position is controlled follows the same method as outlined in previous chapters which is to vary the relative positions of

the gripper and the path of the object, i.e. the track in this context. This is shown in Figure 5.14

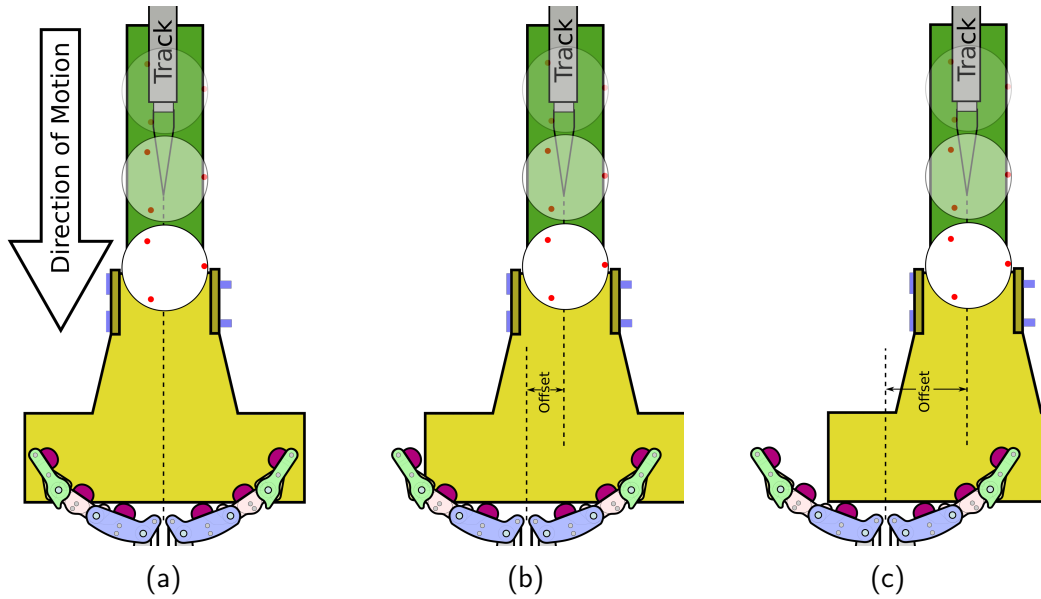


Figure 5.14: Experimental apparatus focused on how positional errors in the object interception are introduced. (a) No error , (b) Small induced error (c) Large induced error

5.2.4 Controlling the grasp initiation time

The highly controlled nature of the path of the object in this experiment allows a light gate to be used to monitor the position of the ball as it approaches the gripper. A light gate consists of an IR emitter and receiver pair. This is placed perpendicular to the known path of the object and is triggered when the ball prevents the light signal leaving the emitter from reaching the receiver, see Figure 5.15. This method provides a precise estimation of the position of the ball at a specific point as it approaches the gripper, while maintaining a lower experimental complexity compared to a vision-based object tracking system which is more common in the literature. From this sensor input, testing could be conducted at a range of grasp timings, implemented with software delays between when the ball triggers the light gate and initiating the grasp. This method, combined with knowledge of the test speed, can be used to trigger the grasp as a function of distance from gripper (as is common in prior examples), time to collision or any other function which is found useful for determining the time to initiate the grasp.

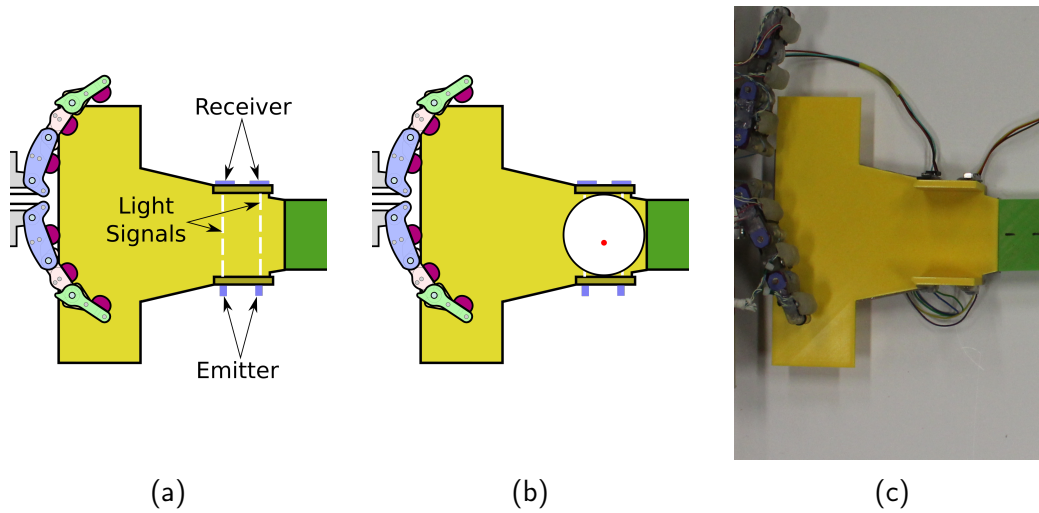


Figure 5.15: Experimental Rig, focused on the light gate (a) Illustration of the light gate, (b) Illustration showing the ball breaking the light gate (c) Photograph of the light gate on the real world apparatus

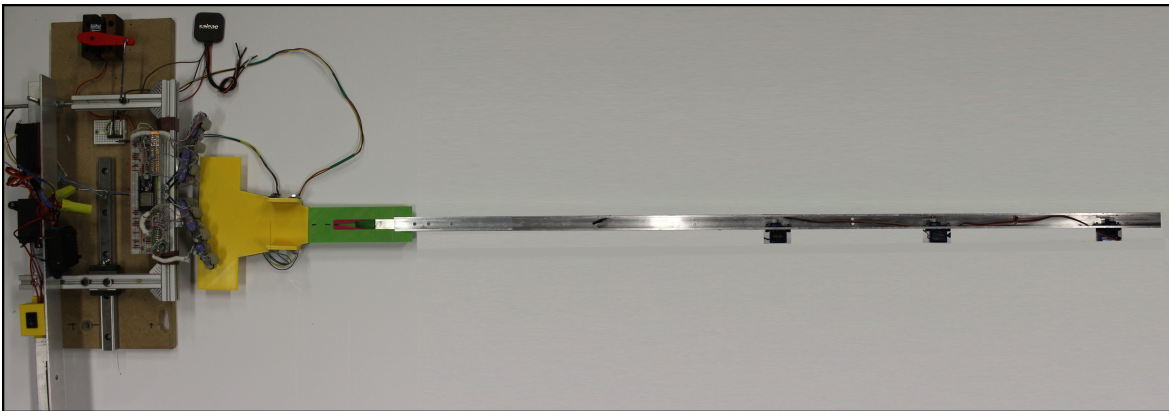
5.3 Summary of real-world gripper and apparatus design

To summarise, the experimental design used for testing during this research, tasks a two-finger gripper with grasping a ball rolling on a horizontal plane under a range of grasping conditions. The experimental apparatus provides mechanisms to systematically vary the object speed, position of the initial object-gripper contact and the grasp timing. A range of control strategies are deployed on an ESP32 microcontroller and their performances are compared by experimentally determining their grasping success rates. The holistic experimental apparatus is shown in Figure 5.16 as both an illustration and a photograph of the apparatus used.

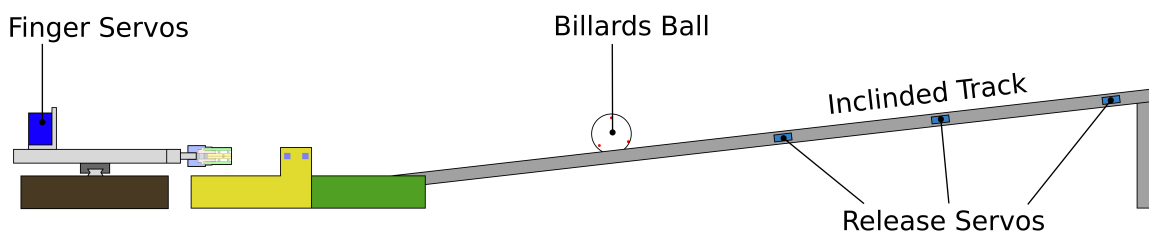
The design of the gripper focuses on simplicity, repeatability and robustness. A two-fingered gripper is chosen as the gripper morphology. The dimensions of each finger are based loosely on the dimensions of the average adult male finger. This both makes its more suitable for human-centric environments and increases the relevancy for the results to a larger range of existing robotic grippers, these dimensions are shown in Figure 5.3. Each finger has three DoF, and underactuation is leveraged to enable one actuator to

control the 3 DoF of one finger. An additional DoF and actuator allow the gripper to move laterally, which results in a gripper with seven DoF controlled by three actuators. The gripper has two types of sensing, joint position sensing and tactile sensing, and uses an ESP32 as the computational unit to read sensors, implement the control strategy and control the gripper's actuators. Drawings of the final gripper design from multiple perspectives is shown in Figure 5.17.

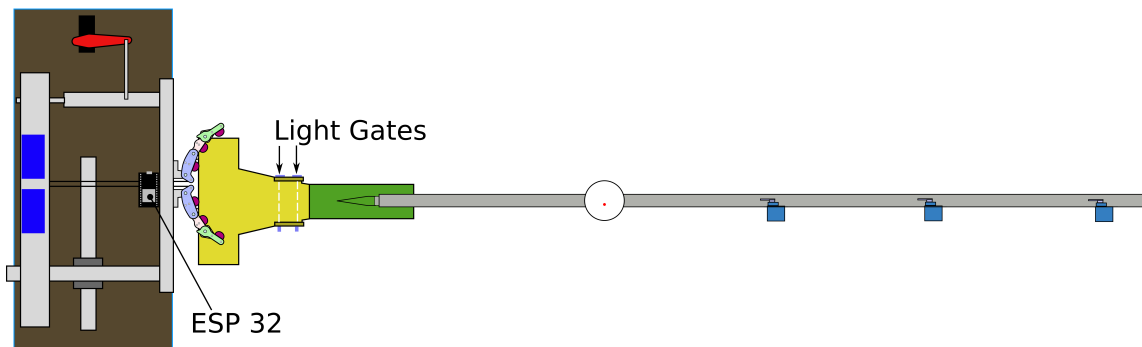
The research presented in this chapter was published in MDPI Sensors. A video illustrating the real-world experimental apparatus was submitted as supplementary material for that submission and is available at: https://drive.google.com/drive/folders/1kJvliWWeTIQRF-fGztb1bo-6VYG_wLpq?usp=sharing.



(a)



(b)



(c)

Figure 5.16: Experimental apparatus, (a) Photograph of the experimental apparatus from above, (b) An elevation view diagram of the experimental apparatus (c) A plan view diagram

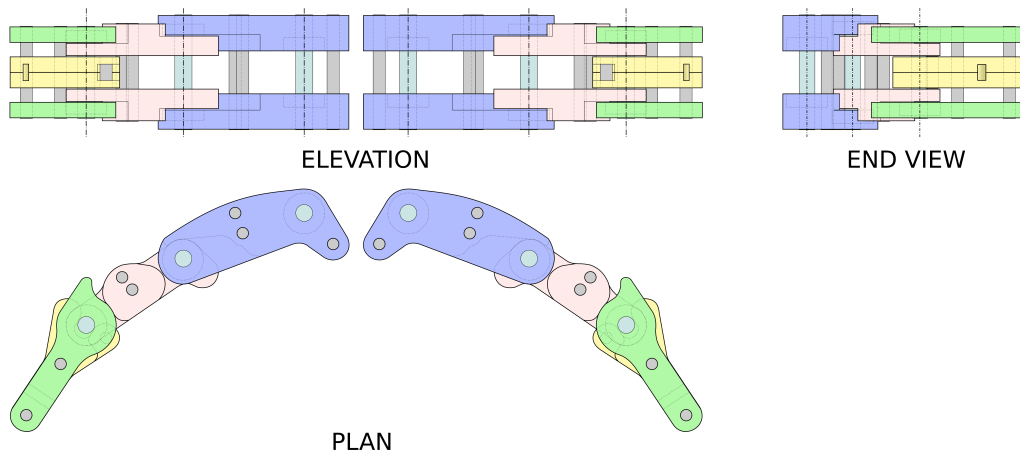


Figure 5.17: Drawing of the final gripper design from three perspectives. Plan, elevation and end view

5.4 Determining the optimum grasp timing

The control strategy used in examples from the prior art is a predictive strategy, where attempted grasp is based purely on an estimation of an appropriate position and time to intercept the object. In order to test this strategy there must first be a definition of the optimum time to initiate the grasp for the predictive strategy so that temporal offsets from that optimum can be induced thereby assessing a strategies robustness to temporal errors.

The majority of examples taken from the prior art initiate the grasping motion when the object passes a threshold distance from the gripper [23, 28, 29, 30, 31]. In these examples, this threshold distance is determined empirically by modifying the threshold distance until the desired gripper performance is achieved. This process is mirrored in this research, to determine the optimum time to initiate the grasp for the experimental setup described earlier in the chapter. A set of tests were conducted, where the gripper initiated the grasp immediately after the light gate was broken. The position of the light gate was then modified until the gripper demonstrated the desired behaviour. It was found that the light gate positioned 58mm from the gripper resulted in high grasping performance at all test speeds.

It is worth noting at this point, that the frame of reference used to define the optimum

grasp initiation time has changed relative to that used during the initial simulated testing. The simulated testing outlined in Chapter 4 defined the optimum grasp initiation time as some time before contact between gripper and object, the real-world testing will use the relative positions of the gripper and object and define the optimum as the time when the object is some threshold distance from the gripper. This change is driven by two main factors. First, using a threshold distance to trigger the grasping motion is more common in existing literature. Second, simulation allowed a large number of tests to be conducted and an exhaustive search of potential grasp timings to be explored, the change in the reference frame reflects the challenges associated with conducting the same exhaustive search in a real-world environment.

5.5 Experimental procedure

Having developed a real-world experimental apparatus, and determined the optimum time to initiate the grasp, a set of tests are conducted to evaluate the performance of each control strategy under a range of grasping conditions.

The procedure for each test begins with the ball being placed on the inclined track and held in place by one of the release servos. On command, the servo releases the ball and it begins to accelerate down the track. The ball then transitions to the flat plane, passes through the light gates and where the gripper attempts to grasp it. After the attempt, the platform on which the ball was rolling is lowered to remove any support force provided by the ground. The grasp was deemed a success if the gripper was able to support the full weight of the ball for 5 seconds. This experimental procedure is illustrated in Figure 5.18.

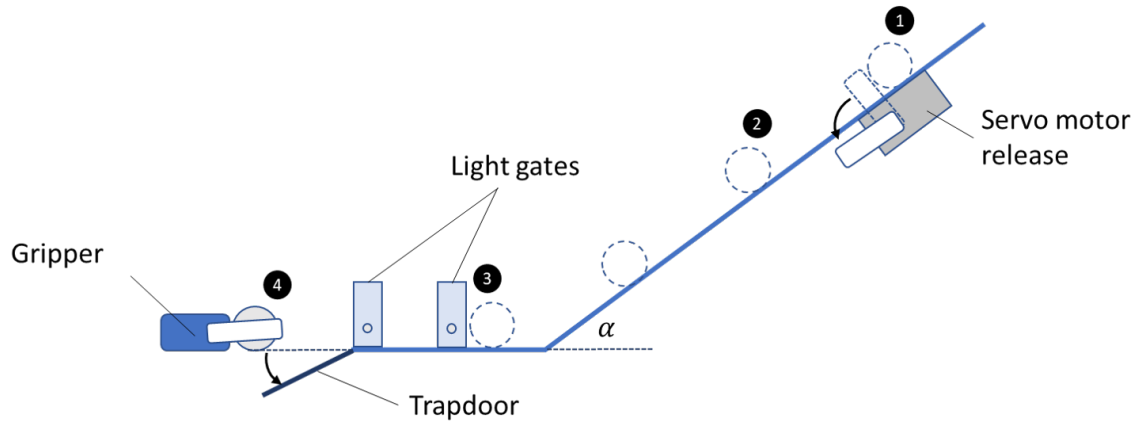


Figure 5.18: Illustration of the experimental procedure. The ball is set in motion by the release of a servo arm (1). The ball rolls down the incline (2) and through the light gates which records its velocity and notifies the control strategy (3). The gripper grasps (or attempts to grasp) the ball and the trapdoor is released to determine if the full mass of the ball can be supported by the gripper (4)

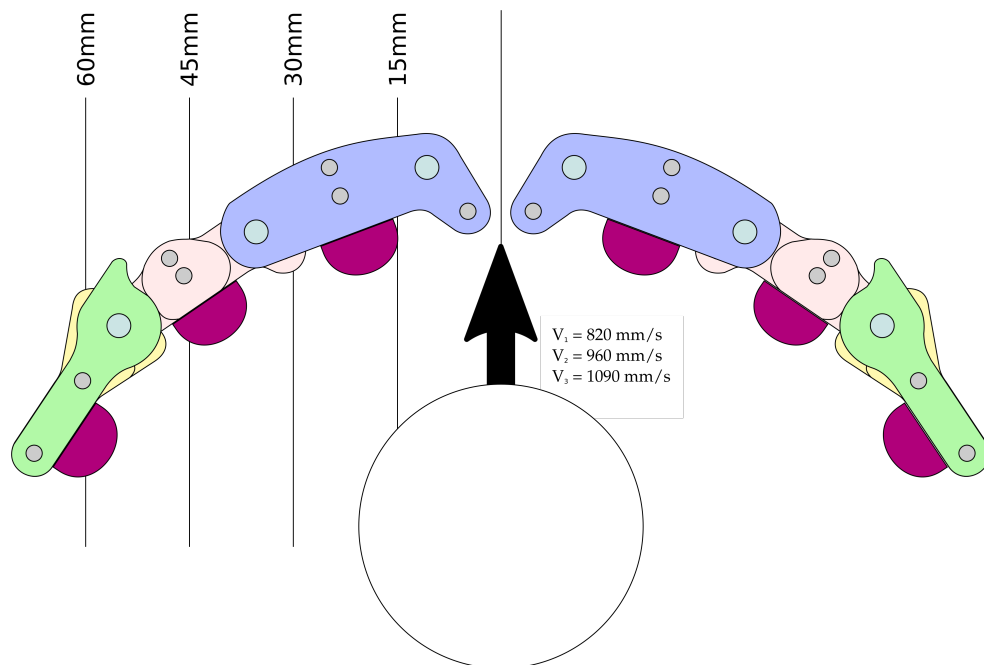


Figure 5.19: Tests were performed at 3 ball speeds and 5 spatial offsets as indicated in this graph.

5.6 Results

A series of experiments were conducted to compare the performance of the reactive and predictive control strategies. The performance of each strategy was measured at three object speeds (0.82m/s, 0.96m/s and 1.09m/s) and at five finger-object contact points (0mm, 15mm, 30mm, 45mm, 60mm), see Figure 5.19. The range of object speeds and finger-object contact points, where chosen as a range which represents conditions under which the predictive grasping strategy can both consistently grasp the ball, i.e. 0.82m/s and 0mm, and conditions where the performance of the predictive strategy deteriorates, i.e. 1.09m/s and 60mm. By examining across this range, the value of the reactive strategy can be determined. The performance of the predictive strategy was quantified using three grasp initiation times. One immediately after the light gate is triggered, previously determined to be optimal. One 5ms and another 10ms after this empirically determined optimum. Ten tests were performed for each set of conditions with the dependent variable being if the grasp was successful or not.

The success rates of the predictive grasping strategy, for the three grasp timings, and the reactive grasping strategy is shown in Figure 5.20.

To examine controller performance across all test conditions, data from all experiments (i.e. different ball speeds and gripper-object contact points) was aggregated for each control strategy (Table 5.2). From 150 grasping trials, the reactive controller that used tactile sensing achieved the best performance (79.3% success rate) followed by the predictive controller that initiated the grasp immediately after passing the light gate (71.3% success rate), 5ms after passing the light gate (70.6% success rate), and 10ms after passing the light gate (51.3% success rate).

A chi-square test was performed to examine the results. These results are presented in Table 5.4. A 95% confidence interval was calculated to compare the reactive grasping strategy to the predictive strategy at three different grasp timings, these confidence intervals are presented in Table 5.3.

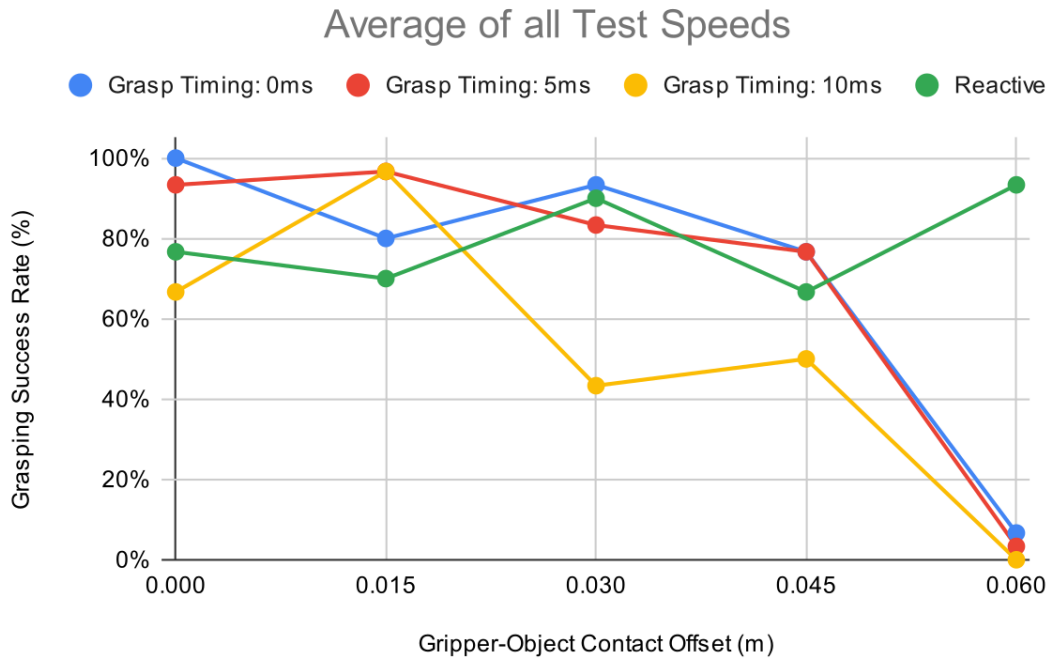


Figure 5.20: Grasping Success Rate Vs Spatial Offset, Averaged Across all Test Speeds

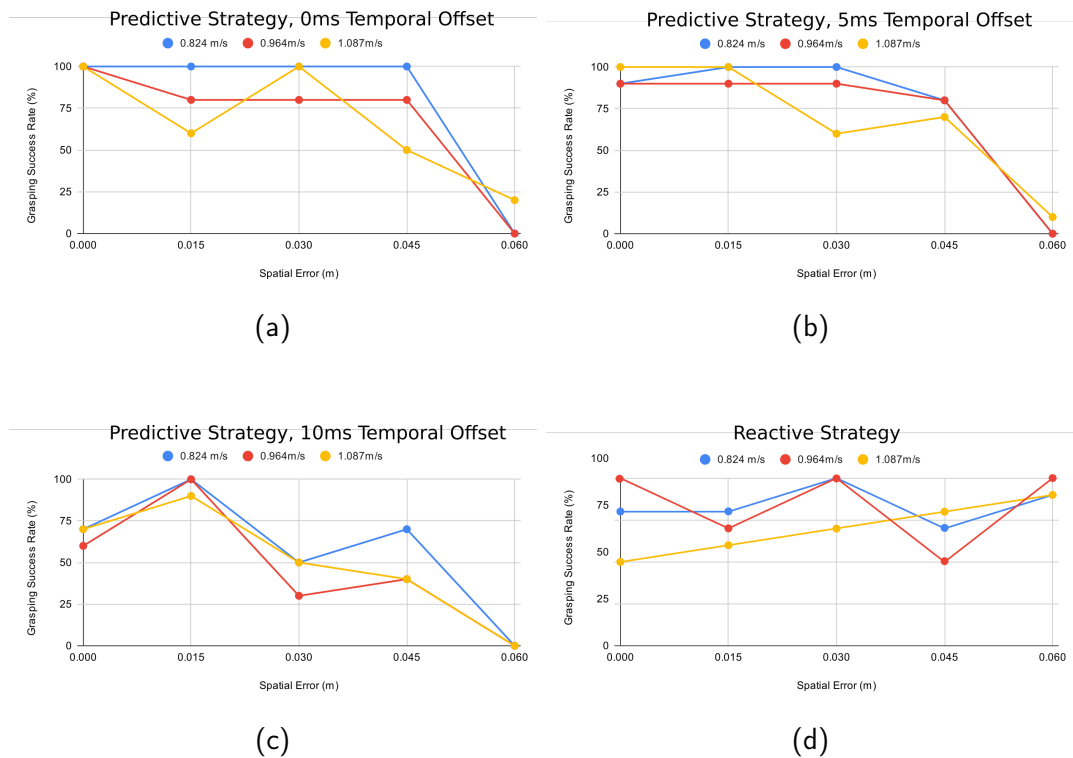


Figure 5.21: Successful grasp rates, observed over 10 tests under each set of conditions. (a) Predictive control strategy with 0ms temporal offset (b) Predictive control strategy with 5ms temporal offset (c) Predictive control strategy with 10ms temporal offset (d) Reactive control strategy

Table 5.2: Summary of grasp performance across all test conditions for each control strategy, where ‘Delay’ is the delay after passing the light gate before the grasp is initiated for the ‘Predictive’ condition,

Delay	Number of Successful Grasps		χ^2	p-value
	Predictive	Reactive		
0ms	107		2.171	0.141
5ms	106	119	2.560	0.110
10ms	77		24.740	0.000

Table 5.3: 95% confidence intervals, comparing the performance of the reactive grasping strategy with that of the predictive strategy at 0ms, 5ms, and 10ms temporal offsets.

Grasp Timing	Confidence Intervals	
	Lower Limit	Upper Limit
0ms	0.98	1.27
5ms	0.98	1.28
10ms	1.29	1.84

Table 5.4: Summary of results from grasping experiments comparing a reactive strategy with a predictive strategy. Where ‘Spatial Offset’ is the distance from the centre of the gripper to the point on the finger where the object makes first contact, and ‘Grasp Timing’ is the delay after passing the light gate before the grasp is initiated for the ‘Predictive’ condition. An asterisk indicates a significant difference in performance at a significance level (α) < 0.05

Spatial Offset	Speed	Grasp Timing	Number of Successful Grasps		χ^2	p-value
			Predictive	Reactive		
0.00m	0.82m/s	0ms	10	8	0.5556	0.4560
		5ms	9		0.0000	1.0000
		10ms	7		0.0000	1.0000
	0.96m/s	0ms	10	10	0.0000	1.0000
		5ms	9		0.0000	1.0000
		10ms	6		2.8125	0.0935
	1.09m/s	0ms	10	5	4.2667	0.0389*
		5ms	10		4.2667	0.0389*
		10ms	7		0.2083	0.6481
0.015m	0.82m/s	0ms	10	8	0.5556	0.4560
		5ms	10		0.5556	0.4560
		10ms	10		0.5556	0.4560
	0.96m/s	0ms	8	7	0.0000	1.0000
		5ms	9		0.3125	0.5762
		10ms	10		1.5686	0.2104
	1.09m/s	0ms	6	6	0.0000	1.0000
		5ms	10		2.8125	0.0935
		10ms	9		1.0667	0.3017
0.030m	0.82m/s	0ms	10	10	0.0000	1.0000
		5ms	10		0.0000	1.0000
		10ms	5		4.2667	0.0389*
	0.96m/s	0ms	8	10	0.5556	0.4560
		5ms	9		0.0000	1.0000
		10ms	3		7.9121	0.0049*
	1.09m/s	0ms	10	7	1.5686	0.2104
		5ms	6		0.0000	1.0000
		10ms	5		0.2083	0.6481
0.045m	0.82m/s	0ms	10	7	1.5686	0.2104
		5ms	8		0.0000	1.0000
		10ms	7		0.0000	1.0000
	0.96m/s	0ms	8	5	0.8791	0.3484
		5ms	8		0.8791	0.3484
		10ms	4		0.0000	1.0000
	1.09m/s	0ms	5	8	0.8791	0.3484
		5ms	7		0.0000	1.0000
		10ms	4		1.8750	0.1709
0.060m	0.82m/s	0ms	0	9	12.9293	0.0003*
		5ms	0		12.9293	0.0003*
		10ms	0		12.9293	0.0003*
	0.96m/s	0ms	0	10	16.2000	0.0001*
		5ms	0		16.2000	0.0001*
		10ms	0		16.2000	0.0001*
	1.09m/s	0ms	2	9	7.2727	0.0070*
		5ms	1		9.8000	0.0017*
		10ms	0		12.9293	0.0003*

5.7 Discussion

Results from the experiment show, the strategy which reacted to real-time, tactile sensor data, achieved the highest grasping success rate, with a success rate of nearly 80%. The predictive strategy achieved a success rate of 71.3%, when its grasp timing was optimum, and 70.6%, and 51.3% when its grasp timing was delayed by 5ms and 10ms respectively. Furthermore, the confidence intervals state with 95% certainty, that the odds of achieving a successful grasp using the reactive grasping strategy is 0.98 to 1.27 times higher than when using the predictive strategy with optimum grasp timing, between 0.98 and 1.28 times higher than with the predictive strategy with a 5ms grasp timing error, and finally between 1.29 and 1.84 times higher than with the predictive strategy with a 10ms grasp timing error.

The reactive controller was found to offer the greatest advantage when the gripper-object contact point was further from the centre of the gripper, where it significantly outperformed the other gripping strategies ($p < 0.05$), see Table 5.4. These findings support earlier results obtained through performing a series of equivalent tests in simulation, outlined in Chapter 4. Furthermore, this verifies our original hypothesis that the implementation of a reactive approach that uses tactile sensor data can enable a more adaptive grasp that is more robust to errors in the position of the initial gripper-object contact.

Results, shown in Figure 5.20, show that the predictive strategies performed best when the point of initial gripper-object contact is near the centre of the gripper (the optimum position). Performance deteriorated as the gripper-object contact moved towards the extremity of the finger. These findings are consistent with results simulation-based tests from Chapter 4 and confirm our hypothesis that an adaptive grasping motion, based on real-time tactile sensor feedback, could mitigate the effects of errors in the interception of a ball by a robotic gripper, resulting in a more robust control strategy for grasping moving objects.

Findings from the predictive controller suggest that the best grasp performance was achieved when the grasp initiated as soon as the object entered its grasp envelope, i.e. immediately after triggering the light gate. This was especially true at higher object speeds, where a delayed grasp response often led to the ball bouncing off the finger on contact and subsequently escaping from the grasp. Since the reactive controller only initiated on contact with the tactile sensor, it is likely that performance improvements in this control strategy were not due to optimization of grasp initiation, but were attributable to other factors of the grasp behaviour, namely the heuristics outlined in Section 3.4.2. These are the lateral movement which centres the object in the gripper (heuristic 2) and the delay induced in the closing motion of the finger that made contact with the object (heuristic 3). This is supported by the data because the reactive grasping strategy was found to perform best when the ball contacted the fingertips, where the effect of these two behaviours was large, and worse closer to the centre of the gripper, where the behaviour of the system approximated a predictive controller with a longer grasp initiation delay.

Sub-optimal grasp initiation may not have been the only factor that affected the performance of the reactive controller. The low spatial resolution of the tactile sensing, i.e. only three tactile sensors per finger, is also a limitation of this implementation. This implementation associates each tactile sensor with a grasping motion, which acts to grasp an object in contact with that sensor. The response triggered by each sensor must be effective at the range of gripper-object contact positions which trigger that sensor. A gripper with higher density tactile sensing could provide a larger range of available adaptive motions for more specific spatial errors.

The results of testing in the real-world, in many ways confirm the findings of simulated testing outlined in Chapter 4. Both find that the average grasping success rate across all test conditions is higher for the reactive grasping strategy compared to the predictive grasping strategy. Furthermore, both highlight the ineffectiveness of the predictive strategy in the face of interception errors. Finally, both testing modalities show that the predictive grasping strategy matches and can outperform the reactive strategy at low

offsets, both spatial and temporal, demonstrating that the choice of grasping strategy is context dependant. In a situation where the robot has a high level of certainty regarding state of the target object, and a highly accurate interception of the object is likely, a predictive strategy is more appropriate. This suggests that a hybrid strategy could be an effective way to maximise the probability of achieving a successful grasp.

There are several differences which can be observed between the two sets of test results. Most notable among these is the effect of spatial offsets on the reactive grasping strategy. In simulated testing as the contact point between gripper and object moves away from the center of the gripper, the performance of the reactive strategy dips briefly at small spatial offsets (this is discussed in Section 4.6), then falls consistently until the object is out of the gripper range. This observed relationship is more complex in the results of real-world testing, where the highest performance was achieved when the gripper-object contact occurred at 60mm offset from the center of the gripper. This highlights the importance of the behaviour of the ball after contact, i.e. the rebound. The nature of the rebound had a huge impact on the success rate achieved by the reactive strategy for both sets of testing and was highly dependant on the induced spatial offset and the dynamic behaviour of the gripper during contact. The complexity of this interaction makes the exact behaviour of any particular real-world system, very difficult to precisely replicate in simulation, resulting in the observed difference in the results. This highlights the need for more research examining the nature of the rebound, including how it might be structured to assist with grasping a moving object.

The test results presented here illustrate that the reactive grasping strategy reduces the spatial precision required to grasp the ball by 15mm, compared to the predictive grasping strategy at the lowest speed and with optimal grasp timing. This performance improvement is even greater at higher speeds and when there is induced temporal errors. The magnitude of this improvement is comparable to that achieved by research conducted on the Comau Smart-Six robot manipulator endowed with a custom robotic gripper. In this example, an improvement in the interception accuracy of 20-30mm is achieved through a novel technique to combat sensor noise. Similarly, research conducted on the

Rollin Justin platform specifies that a spatial resolution of 20mm was required in order to achieve a successful grasp on the Rollin Justin platform. The required resolution varies from platform to platform and, the results of testing presented in this chapter show relatively high grasping success rates on this robotic gripper, up to an offset of 45mm, though this is lower for higher speeds. When using a reactive grasping strategy, the gripper used in this research is able to consistently grasp the object at an offset of 60mm. These results suggest that an adaptive grasping motion has the potential to enable similar reductions in the required spatial resolution on platforms like the Rollin Justin platform. Likewise, the required temporal resolution to achieve a successful grasp is cited as 5ms for the Rollin Justin platform. The results of testing show that for this system, a 5ms temporal error causes a very minor performance reduction ($< 1\%$) however a 10ms temporal error causes the performance to drop by 20%. A reactive grasping strategy eliminates the problem of temporal errors, since the grasp is triggered by contact and could enable the Rollin Justin platform to achieve a successful grasp, even when an accurate estimate of the interception is not possible.

The advantages outlined here result in a grasping strategy which is more robust to errors in a gripper's interception of a moving object. Leveraging these advantages means a successful grasp can be achieved with a less accurate estimated interception, helping to tackle common problems such as occlusion, computational limitations and lighting conditions.

In this chapter, the benefits of incorporating tactile feedback in the grasping strategy was evident. However, this research also serves to highlight several avenues for future research. The heuristic based use of the tactile sensing, although providing improved performance, is not optimised. Machine learning techniques could improve upon the performance and are investigated in Chapter 6. The testing outlined here used relatively low tactile sensor density, with just six tactile sensors used. Increased tactile sensor density could provide more accurate information about the location of the object and enable a more optimal reactive grasping motion.

6 | Improvement through Machine Learning Techniques

Previous chapters have proposed a heuristic-based strategy which creates reactive grasping behaviour, and successfully demonstrated increased robustness when grasping a moving object, both in simulation and using a real-world gripper. Machine learning is a technique which has been growing in popularity in recent years and has been shown to be particularly effective in similarly complex applications. This chapter explores the potential of machine learning techniques, to train a neural network how to perform adaptive grasping for objects in motion. It aims to design a machine learning agent, train the agent in simulation, then deploy and evaluate its performance on a real-world gripper, using the experimental methodology outlined in Chapter 3 Section 3.1. A neural network is trained, first using synthetic data and then in simulation, before being adapted for use on a real-world gripper. Tests are conducted to quantify the performance of the neural network at a range of test speeds and gripper-object contact points using a real-world gripper. This is then compared to the results of testing conducted on a traditional, predictive strategy and the heuristic based, reactive strategy outlined, outlined Chapter 5 Section 5.6.

6.1 Machine learning agent

6.1.1 Choosing a neural network

A feed forward, deep, neural network (DNN) is chosen as the agent, which will be used by the control strategy tested this chapter. There are several reasons why a neural network agent is suitable in this application. Neural networks are:

- **Efficient at runtime.** Neural networks learn how to achieve desirable outcomes through training. This training is an iterative process of modifying the network's weights to modify its behaviour. This process requires a large number of training cycles and therefore long training times. However, once these weights have been trained, executing the network is simply a series of matrix multiplications, as the input values are passed through the network. Matrix multiplication is relatively easy for a microcontroller, has already been highly optimised and is suitable for parallelisation. Therefore, despite long training times, the computational overhead of a neural network at runtime is low. This is particularly important for a controller grasping a moving object since reducing latency is of the upmost importance and robots are often computationally constrained devices.
- **Capable of learning complex behaviour.** Trained, DNNs are capable of learning and modeling complex relationships between inputs and outputs. Significantly more complexity can be achieved with similar or fewer parameters using a neural network in comparison to alternative agents such as decision trees, kNN, SVM, etc. The ability to model more complexity with fewer parameters enables better performance while requiring less computational power and memory resources. The ratio of computational power required, to the resulting behavioural complexity, makes it suitable for robotics, and grasping moving objects in particular.
- **Modular and reconfigurable** Neural networks have been shown to be highly modular and reconfigurable. Training conducted for one application can be utilised for a similar application by retraining the network. This can involve retraining the

entire network, retraining a subset of the layers, or adding layers. Since training an agent is highly time consuming and costly, minimising the amount of training required, by starting with an existing network, is advantageous. For example, when conducting future research, the model which is used in this chapter for a two-finger gripper can be adapted to be used in a more complex grasping scenario. It is also useful more generally for robots deployed in the real-world where the strategy which enables them to interact with moving objects need not be independently trained for each gripper arrangement or application. Instead, it can be taken from a more general network, and a smaller number of retraining cycles used to adapt it for a given grasping system or application.

- **Capable of producing both continuous and categorical outputs.** Finally, neural networks have the advantage of readily providing both continuous and categorical outputs, which is not possible with many alternatives. A categorical output is used to specify one of several discrete possible categories, in this case to indicate whether to begin to close the finger or not. In contrast, continuous outputs provide a value which exists in some range of possible values. These are often essential in robotics applications. In this case, continuous outputs are needed to specify the speed and distance of the lateral movement.

6.1.2 Structure of the network used

When choosing a network structure, it was important to carefully determine the required network complexity. The network must be sufficiently complex to create an optimised adaptive grasping strategy, but excessive complexity, will increase training times, increase the number of parameters and therefore memory demands, and increase the computation time for a forward pass. An iterative approach was taken to choosing the network structure, including the number and nature of the inputs, the number of hidden layers, the number of nodes in each layer, and the number and effect of each of the outputs. The process started with a simple structure, with 2 hidden layers, 6 input nodes, and 4 output nodes. This was iterated upon until a network with the desired performance

characteristics was produced. On each iteration, training was conducted using the latest version of the network structure, the progress of training and the resulting performance was examined. This informed what and how additional complexity was included.

The result of this iterative process was a network structure that has 14 input nodes, 4 hidden layers each with between 25 and 50 nodes, and 4 output nodes, this is illustrated in Figure 6.1. Though more complex structures have been demonstrated in the literature [186] and may lead to better performance, this simple structure is sufficient to address the research aims.

The output nodes, shown in Figure 6.2, are mapped to a system which controls the actuators of the gripper. The four output nodes correspond to:

1. A command to close the left finger.
2. A command which closes the right finger.
3. A value which is mapped to the speed of lateral motion, this also controls the direction of motion.
4. A value which is mapped to the absolute magnitude of the lateral motion.

This implementation affords the neural network strategy control of the gripper, which is very similar to that given to the heuristic-based reactive strategy tested in Chapter 5.

The input nodes, shown in Figure 6.3, act as the sole source of information for the neural network strategy regarding the state of the system. The first 6 nodes ($I_1 - I_6$) contain the most recent data from the tactile sensors in the fingers, one input node per sensor. This provides the most up-to-date data regarding the state of the tactile sensors and is essential to the reactive grasping strategy. This tactile sensor data is binary, i.e. contact or no contact.

The next 4 each contain a sum of sensor readings for a finger at some point in the recent past. Initial testing with the neural network showed a significant advantage to providing the neural network information about the state of the tactile sensors in the recent past.

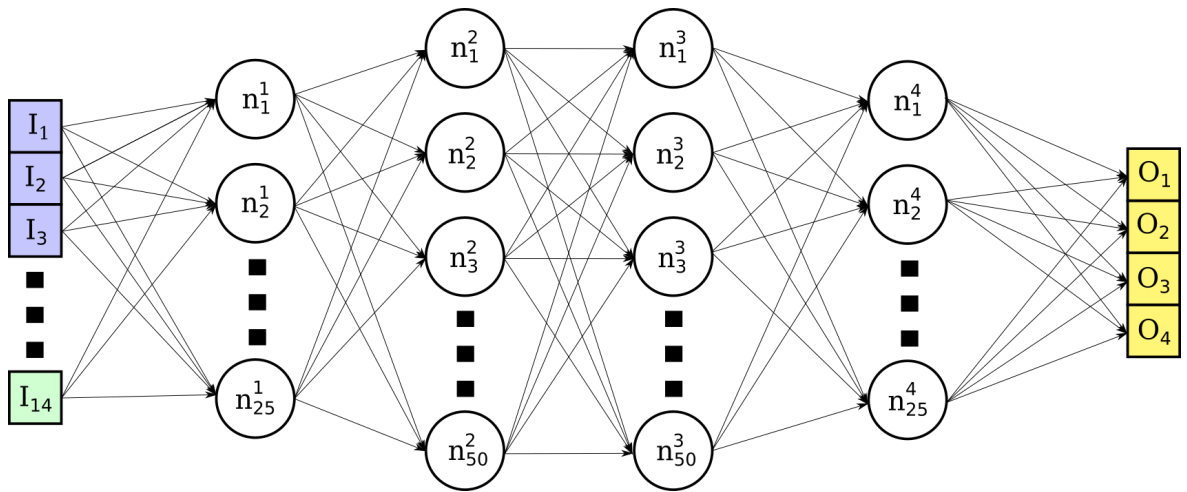


Figure 6.1: Illustration of the structure of the neural network used. 'I' represents the input nodes, 'O' the output nodes. 'n' represents the nodes in the hidden layers with the superscript showing the layer it belongs to and the subscript the node number within that layer.

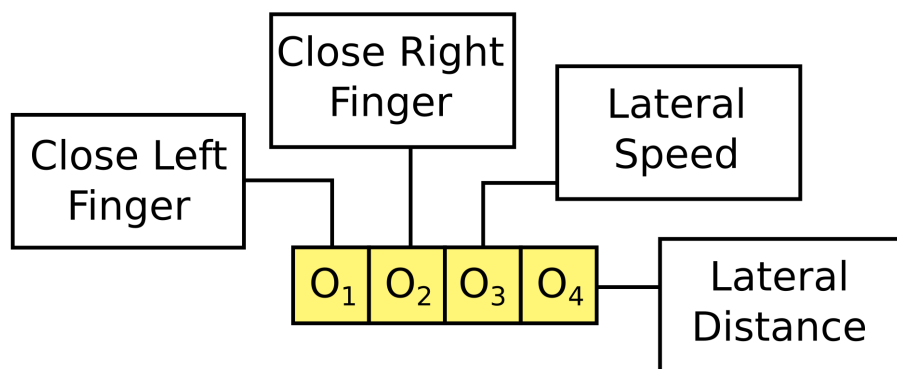


Figure 6.2: Illustration of the output nodes of the neural network used

Several versions of how to encode this information was evaluated. For example, having an input node for each of the tactile sensors at 1 point in the past, resulting in an additional 6 input nodes. Alternatively, including an input node for each of the tactile sensors at 2 points in the past results in an additional 12 input nodes. Ultimately, though there was significant advantages to including information regarding the state of the tactile sensors in the recent past, there was limited benefit to including this information in its entirety, and 12 additional input nodes increases the complexity significantly. Therefore, the network which was ultimately used condensed this information and passed it to the network through 4 additional input nodes. Specifically, node 7 refers to the sum of the sensor readings of the left finger at one time step before the current time ($[I_1 + I_2 + I_3]_{@t-1}$), node 8 contains the right finger at one time step before the current time ($[I_4 + I_5 + I_6]_{@t-1}$), node 9 the left finger at two time steps before the current time ($[I_1 + I_2 + I_3]_{@t-2}$) and finally node 10 the right finger at two time steps before the current time ($[I_4 + I_5 + I_6]_{@t-2}$).

The final 4 input nodes correspond to a history of the actions previously conducted. In a similar way to the historic values of the tactile sensors, a history of the completed actions was shown to improve both training times and eventual performance during preliminary testing. Since this information is freely available to the robot's computational system, and does not increase memory or computational demands significantly, this information was included as the final 4 input nodes. The first node indicates the current state, i.e. closed or open, of the left finger (node 11), then the right finger (node 12) and finally recording that the gripper moved laterally to the left (node 13) or right (node 14).

The choice of input nodes explored here is just one of many feasible alternatives. Many of the choices made used anecdotal evidence, had limited testing and often required using best judgment. The inputs chosen were sufficient to address the research goals outlined in this thesis, though a rigorous examination of the optimal input structure may conclude that a different set of inputs would yield better performance.

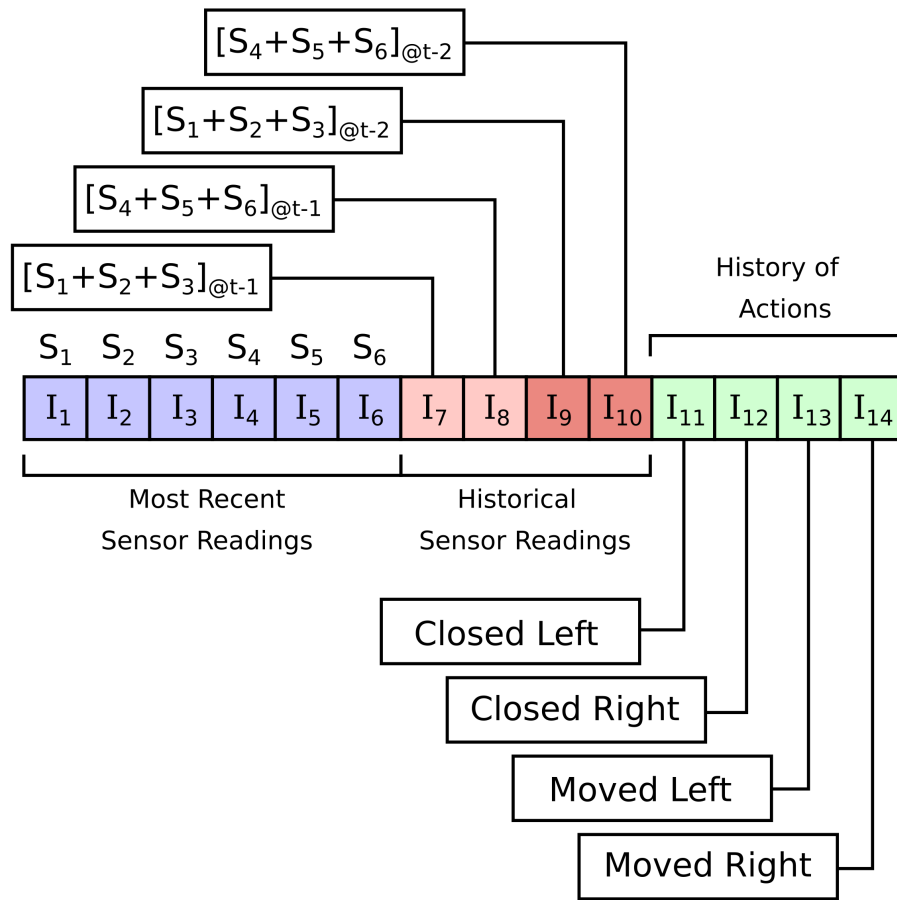


Figure 6.3: Illustration of the input nodes of the neural network used

6.2 Training the agent

Simulation is often used as a tool when applying machine learning to robotic applications. This approach has the major advantage of lowering the cost of both; generating a large amount of data, and executing the large number of training cycles required. This is discussed further in the context of existing literature in Chapter 2.

The neural network training was broken in three stages. First, supervised learning leveraged synthetic data to pretrain the network. Next, reinforcement learning used the simulated testing environment, discussed in Chapter 4, to further improve the network. Finally, the network was adapted for use on the real-world gripper.

6.2.1 Supervised learning

Agents deployed in robotic applications tend to have very large actions spaces, i.e. a large number of possible actions which the agent could take. In this particular application, only a very small subset of this action space will result in a successful grasp. Due to the large size of the action space relative to the small action subspace which results in a successful grasp, it is exceedingly rare to achieve a successful grasp by randomly sampling possible actions. However, reinforcement learning techniques rely on rewarding the network for desirable behaviour, in this case when a successful grasp is achieved. To overcome this sparse reward function, it is necessary to first pretrain the network using supervised learning. This has the effect of initialising the neural network to the relevant region of the action space, greatly increasing the chances of achieving a successful grasp with the pseudo-randomly sampled actions. This dramatically reduces the required number of training cycles required for successful reinforcement learning.

Supervised learning techniques rely on a labeled dataset. A synthetic dataset was created by recording the sensor data during approximately 50,000 simulated gripper-object collisions across a range of spatial offsets and object speeds. After collecting this sensor data, each sensor reading was labeled with the action which the heuristic-based reactive

strategy (outlined in Chapter 3) would have taken under those circumstances. The result of using this dataset to train the network, is behaviour which is similar to that which has already proven effective at creating adaptive grasping motions during testing in Chapters 4 & 5. The network was then trained to minimise the difference between the output values of the network and the ground-truth labels provided by the heuristic-based reactive strategy.

In order to visualise this process, the dataset is discretised into eleven categories based on the first, non-zero tactile sensor reading for that simulated collision. This is a non-complete representation of the training but is sufficient to visualise and monitor the progress of training. Figure 6.4 shows this visualisation, graphing the difference between the output of the network and the datapoint's label for each of the eleven categories for the duration of training.

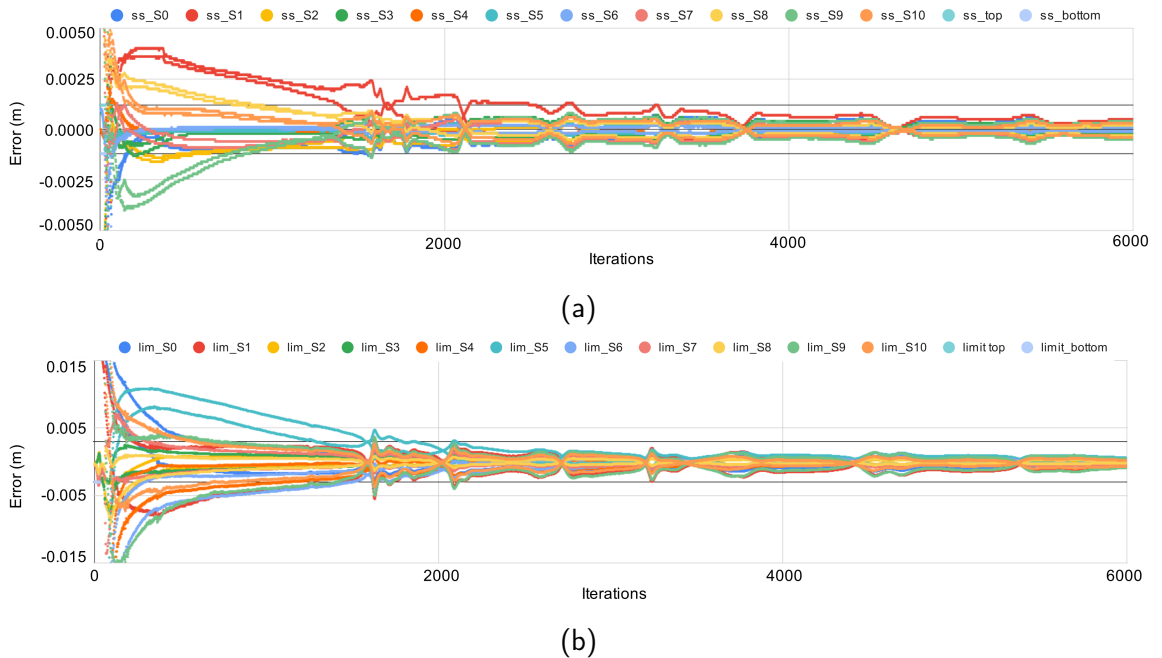


Figure 6.4: Grasp showing the progress of the supervised pretraining, graphing errors over training iterations. **(a)** shows the errors in the network output associated with stepsize of the lateral motion position controller, **(b)** shows the errors in the network output associated with lateral movement limit

6.2.2 Reinforcement learning

The next step taken was to use reinforcement learning (RL) to further optimise the network. This step was conducted in simulation, using a reinforcement learning technique called policy gradients. An agent-in-training takes some input regarding the state of the system, and its output corresponds to some recommended action from the action space. In policy gradients some random deviation is added to the output, thereby modifying the recommended action. This modified action is then executed. This process is repeated until some measurable effect of the actions taken can be observed, i.e. if an attempted grasp is successful or not. The desirability of the outcome is scored according to some reward function. Furthermore, each action taken which ultimately resulted in that outcome is assigned a value describing how influential that action was in determining the outcome. By doing this, responsibility for the outcome is attributed to some actions more than others. By scoring the ultimate outcome and attributing the result to the actions taken, each random deviation added to the network output can be determined to be a deviation leading to a more favorable outcome or less favorable outcome. In the case of training a neural network agent, back propagation is used to modify the networks weights such that positive outcomes are more likely and negative outcomes less likely to occur in the future.

Adaption of the simulated environment to allow training

In order to facilitate training in simulation, the simulated grasping apparatus and environment used previously in Chapter 4 was adapted to enable it to be used as a simulated training platform.

The policy gradient training was implemented as a ROS node. Training using a ROS node allowed the training process to use the existing infrastructure of ROS nodes to read from the simulated sensors, monitor the state of the simulation and actuate the simulated gripper. In addition to the aforementioned sensing, actuation, and monitoring nodes, the supervisor ROS node was adapted to aid in the training of the neural network. One such

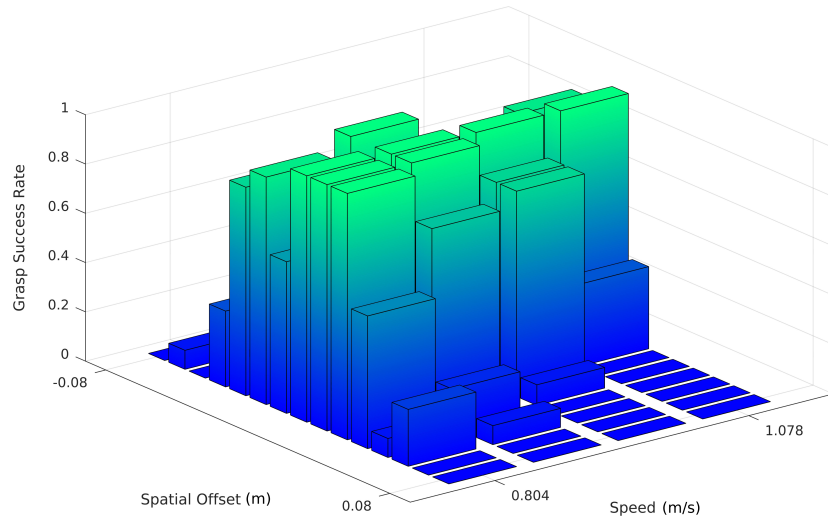
adaption was the way in which the supervisor node varies the grasping conditions from test to test. In previous experiments the supervisor node was tasked with systematically varying the grasping conditions from test to test to ensure the performance of the gripper was evaluated under a range of grasping conditions. When training, this was adapted to consider the current performance of the neural network. By accounting for the current performance of the network the supervisor node can force training to focus on grasping conditions in which the network struggles to achieve a successful grasp. Thereby improving the network's performance across the entire range of grasping conditions tested.

Designing a reward function

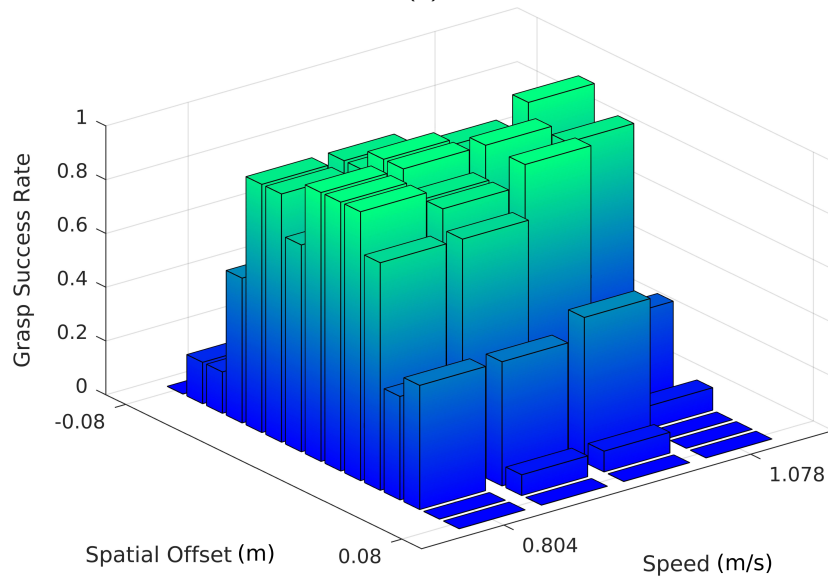
An important factor to consider when using RL is the design of the reward function. This is the way in which desirability of a particular outcome is quantified. Since this quantity is directly responsible for determining how the network is shaped, it is essential to give this careful consideration. Poorly designed reward functions have been shown to yield substandard performance by getting stuck in local optima [217].

In this research there were four different rewards, a weighted combination of which made up the reward function. These were the reward for successfully closing both fingers, moving laterally in the correct direction, failing to grasp the ball and finally successfully grasping the ball. These values were set in a parameter file and multiple different iterations of these attempted during training. The values which were found to work best were $+0.1$ for closing both fingers during the attempted grasp, $+0.5$ for moving laterally in the correct direction, -0.01 for failing to grasp the ball and $+1$ for a successful grasp.

To gauge progress, the performance of the control strategy is assessed in simulation at various stages throughout the training. Figure 6.5 shows this assessment at two major milestones in the training process. First before applying RL, when the network has been initialised by pretraining using supervised learning. Finally, after RL when the network is ready to be deployed on the real-world gripper.



(a)



(b)

Figure 6.5: Plots which visualise the performance at difference stages during the training. (a) shows the performance of the pretrained network before RL is used, (b) shows the performance of the same network after RL is used to train it.

6.2.3 Adaption for use in the real-world

Finally, the network was adapted for use on the real-world gripper. This required the creation of a bespoke application which would manage the upload and execution of a neural network on the real-world gripper's sole computational device, an ESP32. The upload was achieved by writing the learned weights from the computer which ran the simulated training, to the microcontroller which would execute the network and control the gripper during testing. It was essential to ensure data integrity while uploading the weights to the microcontroller, this was achieved by hashing the weight values before and after upload and cross checking that the hash values matched.

During execution of the network on the microcontroller there were two network adaptations which facilitated the deployment of a network trained in simulation for deployment on a real-world gripper. First, the polling frequency of the neural network controlling the real-world gripper was matched to that of the neural network controlling the simulation during training, this was 150Hz. Finally, the outputs controlling the gripper's lateral motion were scaled linearly and a maximum limit set. This was to account small differences between the simulated and real-world grippers, including the initial position of the tactile sensors and smaller range of lateral motion of the real-world gripper.

6.3 Deploying the neural network control strategy on the ESP

A custom application was developed to allow a neural network to be uploaded and executed on an ESP32 micro-controller. This in-turn controlled the real-world gripper, as in the previous testing.

6.3.1 ML control strategy implementation

The learnt control strategy was implemented in the same way as its predecessors, the predictive and reactive control strategies, i.e. as a precompiled application which was flashed to the microcontroller. This application was also created using the ESP-IDF and leveraged the same task-based development approach. In fact, it used many of the same ESP tasks previously outlined in Chapter 5.

The tasks required to implement the ML control strategy were those which; read from the tactile sensors, read from the joint position potentiometers, controlled each of the fingers, and controlled the gripper's lateral motion. Furthermore, a the supervisor task was modified to launched a new neural network task, while maintaining its previous functionality of task handling, memory allocation, calibration and test initialisation.

The new neural network task was created to fulfil two main requirements. First to read the network weights from the computer which trained the network via a serial connection and write them to the microcontroller's memory. Second, to take the most recent tactile sensor readings, create an input vector for the network, executed a forward pass through the network, interpret the output vector, and finally use the aforementioned control tasks to actuate the gripper accordingly.

6.4 Experimental procedure

In order to quantify the performance of a neural network approach, the experimental methodology outlined in Chapter 3, Section 3.1 and the bespoke experimental apparatus, outlined in Chapter 5, Sections 5.1 and 5.2 were used. This involved quantifying the performance of a two finger robotic gripper grasping a ball rolling toward it along a flat surface, and providing a systematic way to test each of the parameters which were thought to effect grasp success. This allowed the performance of the machine learning control strategy to be compared to that of previously tested control strategies.

6.5 Results

A series of experiments were conducted to quantify the performance of a learnt grasping strategy that used a neural network to control the grasping motion. The performance was measured for three different object speeds (0.82m/s, 0.96m/s and 1.09m/s) and at nine finger-object contact points, from -60mm in increments of 15mm to +60mm. Twenty tests were performed at each speed for a finger-object contact point of 0mm and ten tests were performed under every other set of conditions.

The success rate achieved by the neural network strategy, relative to object speed and gripper-object contact point, during this testing is shown in Figure 6.6.

The neural network grasping strategy achieves a higher average grasping success rate than any previously tested strategy, under the same grasping conditions, i.e. contact points and object speeds. On average the neural network strategy achieves a 90% grasping success rate, compared to a 79.3% success rate for the reactive strategy and a 71.3% success rate for the predictive controller using the highest performing grasp timing. Chi-squared tests conclude that on average, across all test conditions, there is a significant difference (at a significant level <0.001) between the performance of the neural network strategy and every iteration of the predictive strategy irrespective of grasp timing, see table 6.1.

Furthermore, chi-squared tests comparing the performance of the neural network strategy and the heuristic-based reactive strategy also conclude that there is a significant difference (at a significance level of $\alpha < 0.005$) in their performance, see table 6.2.

A more granular comparison of the neural network strategy and the predictive strategy using the chi-squared test of independence is presented in table 6.3. Finally, a 95% confidence interval was calculated to compare the neural network grasping strategy to both the predictive strategy at three different grasp timings, and the heuristic reactive strategy, these confidence intervals are presented in Table 6.4.

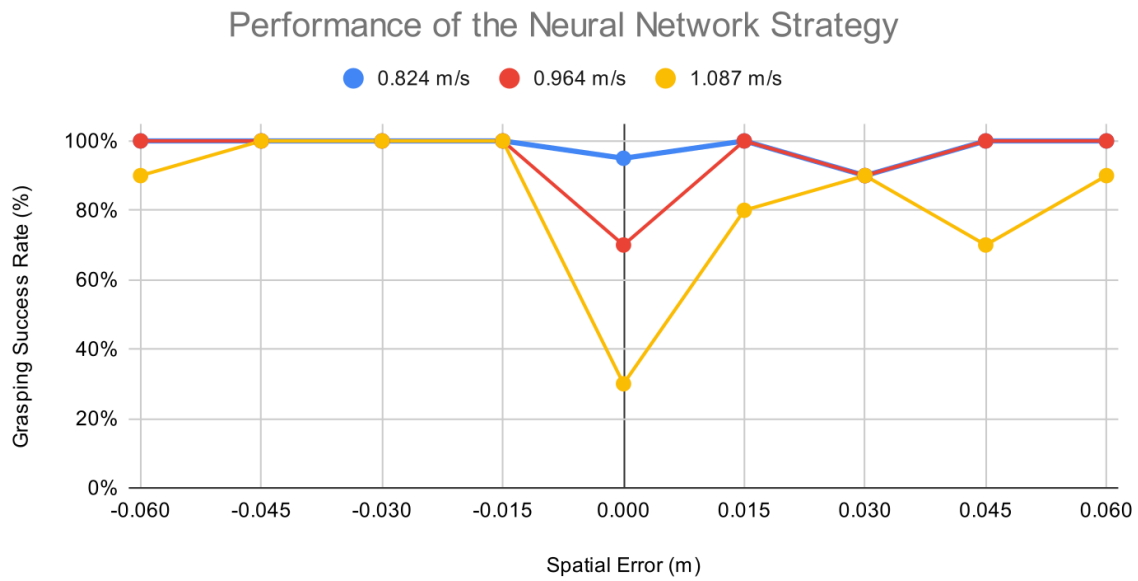


Figure 6.6: Performance of the the Neural Network Strategy at a range of object speeds and gripper-object contact points

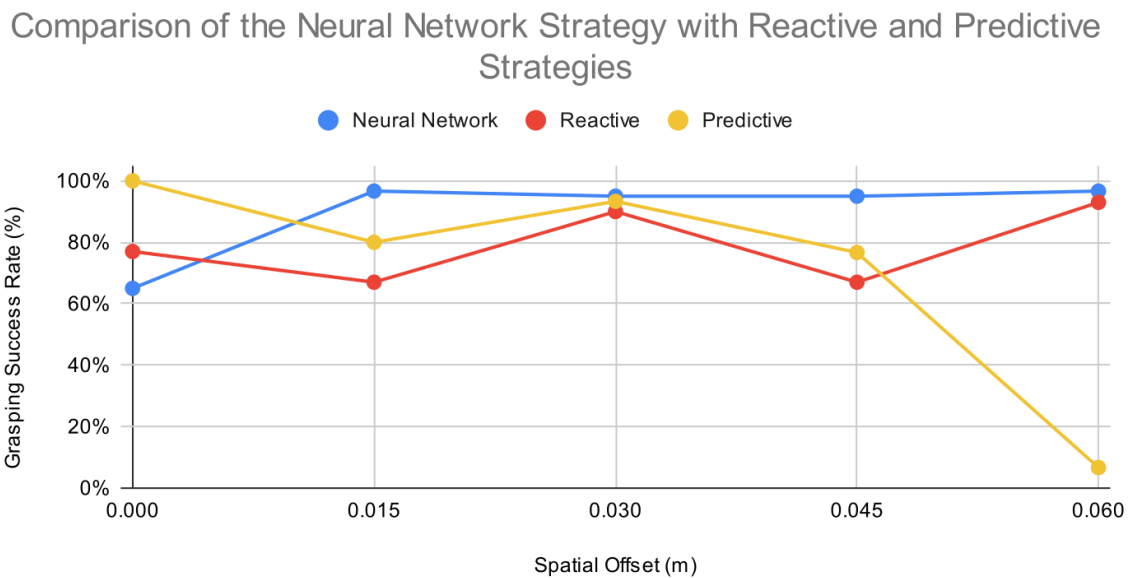


Figure 6.7: Comparison of the performance of a Neural Network strategy with the performance of previously tested strategies, outlined in Chapter 5. The data presented is an average across all test speeds. Furthermore, each data point presented for the neural network's performance is an average of the positive spatial offset with its negative counterpart.

Table 6.1: Comparison of the grasp performance of the Neural Network and Predictive strategies across all test conditions. Where 'Grasp Timing' is the delay after passing the light gate before the grasp is initiated for the 'Predictive' strategy

Grasp Timing	Total Number of Grasps		Number of Successful Grasps		χ^2	p-value
	Predictive	Neural Network	Predictive	Neural Network		
0ms	150	300	107	269	23.1457	0.0000*
5ms	150		106		24.6420	0.0000*
10ms	150		77		80.5499	0.0000*

Table 6.2: Comparison of the grasp performance of the Neural Network and Reactive strategies across all test conditions

Total Number of Grasps		Number of Successful Grasps		χ^2	p-value
Reactive	Neural Network	Reactive	Neural Network		
150	300	119	269	8.1396	0.0043*

Table 6.3: Summary of results from grasping experiments where the gripper is controlled by a neural network. Where ‘Spatial Offset’ is the distance from the centre of the gripper to the point on the finger where the object makes first contact, and ‘Grasp Timing’ is the delay after passing the light gate before the grasp is initiated for the ‘Predictive’ condition. An asterisk indicates a significant difference in performance at a significance level (α) < 0.05

Spatial Offset	Speed	Grasp Timing	Number of Successful Grasps		χ^2	p-value
			Predictive	Neural Network		
0.00m	0.82m/s	0ms	10	19	0.0000	1.0000
		5ms	9		0.0000	1.0000
		10ms	7		1.7668	0.1838
	0.96m/s	0ms	10	14	2.1094	0.1464
		5ms	9		0.5823	0.4454
		10ms	6		0.0188	0.8909
	1.09m/s	0ms	10	6	10.4632	0.0012*
		5ms	10		10.4632	0.0012*
		10ms	7		2.8676	0.0904
0.015m	0.82m/s	0ms	10	20	0.0000	1.0000
		5ms	10		0.0000	1.0000
		10ms	10		0.0000	1.0000
	0.96m/s	0ms	8	20	1.6741	0.1957
		5ms	9		0.1293	0.7192
		10ms	10		0.0000	1.0000
	1.09m/s	0ms	6	18	2.1094	0.1464
		5ms	10		0.0670	0.7958
		10ms	9		0.0000	1.0000
0.030m	0.82m/s	0ms	10	19	0.0000	1.0000
		5ms	10		0.0000	1.0000
		10ms	5		5.8594	0.0155*
	0.96m/s	0ms	8	19	0.4167	0.5186
		5ms	9		0.0000	1.0000
		10ms	3		11.2713	0.0008*
	1.09m/s	0ms	10	19	0.0000	1.0000
		5ms	6		3.6300	0.0567
		10ms	5		5.8594	0.0155*
0.045m	0.82m/s	0ms	10	20	0.0000	1.0000
		5ms	8		1.6741	0.1957
		10ms	7		3.7500	0.0528
	0.96m/s	0ms	8	20	1.6741	0.1957
		5ms	8		1.6741	0.1957
		10ms	4		11.4844	0.0007*
	1.09m/s	0ms	5	17	2.5781	0.1084
		5ms	7		0.2344	0.6283
		10ms	4		4.4643	0.0346*
0.060m	0.82m/s	0ms	0	20	25.6688	0.0000*
		5ms	0		25.6688	0.0000*
		10ms	0		25.6688	0.0000*
	0.96m/s	0ms	0	20	25.6688	0.0000*
		5ms	0		25.6688	0.0000*
		10ms	0		25.6688	0.0000*
	1.09m/s	0ms	2	18	11.7188	0.0006*
		5ms	1		15.0897	0.0001*
		10ms	0		18.9063	0.0000*

Table 6.4: 95% confidence intervals, comparing the performance of the neural network grasping strategy with that of the predictive strategy at 0ms, 5ms, and 10ms temporal offsets, and the heuristic reactive strategy.

	Confidence Intervals	
	Lower Limit	Upper Limit
Predictive 0ms	2.09	5.83
Predictive 5ms	2.16	6.01
Predictive 10ms	5.04	13.44
Reactive	1.31	3.89

6.6 Discussion

Results from the experiment show that the gripper controlled by a trained neural network achieved a higher grasping success rate (90%) than any of the previously tested strategies. In contrast previous strategies achieved an average grasping success rate of 79% and 71% for the reactive strategy and predictive strategy with optimal timing respectively. The magnitude of these improvements was statistically quantified by calculating a confidence interval, shown in Table 6.4. This found that with 95% confidence the odds of a success grasp using the neural network strategy is between 2.09 and 5.83 times higher than with the predictive strategy with optimal grasp timing, and between 1.31 and 3.89 times higher than with the heuristic-based, reactive strategy. The performance difference between the neural network strategy and the predictive strategy was found greatest when the point of contact between the gripper and object was furthest from the centre of the gripper. This was also demonstrated by the chi-squared tests, where p-values of < 0.001 was achieved at all speeds for the maximum spatial offset tested (0.06m). At an initial contact point which is 60mm offset from the center of the gripper it significantly ($\alpha < 0.001$) outperforms all versions of the predictive strategy at all test speeds. This echoes previous findings, from Section 5.6, where the reactive strategy performed significantly better than the predictive strategy when the contact point is at the fingertip.

Results presented show similarities between the neural network strategy and the heuristic-based reactive strategy previously tested. This includes lower performance at contact points close to the center of the gripper and higher performance at higher offsets. The difference between these strategies is the consistency of the neural network strategy which achieved grasping success rates greater than 95% for all non-zero spatial offsets. In contrast, the reactive strategy performance drops $< 70\%$ at 15mm and 45mm offsets. This is reflective of the heuristic based strategy's limitation to using generalised rules to try to react to all contact conditions, unlike the neural network which is trained using prior experience to provide an optimum response for each object state.

A further contribution of the work presented in this chapter is the use of training in

simulation for use on a real world gripper. The high performance of the neural network strategy in the real world, demonstrates the feasibility of training in simulation for robotic control in the real world. In this implementation, high real-world performance was achieved with minimal adaptation and no transfer learning. This was achieved by creating the simulation to replicate the real world as closely as possible. The choice of sensor implementation also facilitated the portability of the neural network. The sensor used was a 3-axis, Hall effect sensor capable of providing continuous tactile information in both shear directions as well as the normal direction. Despite this, the implementation used only force in the normal direction to determine a binary output, i.e. contact or no contact. This made the behaviour of the tactile sensor significantly easier to simulate accurately and reduced the reality gap.

The chosen network structure, a feed-forward network with four hidden layers, is a relatively simple network structure. Improved performance by optimising the structure of neural network, has been demonstrated in previous applications, however it is a significant undertaking and an entire research area of its own. In this research, the chosen network was sufficient to demonstrate the suitability of neural networks and machine learning techniques to this application.

Aside from the choice of network structure, the choice of network output implemented here, though proven effective, is also limited. The chosen outputs creates adaptive behaviour which is very similar to that of the reactive strategy tested in Chapter 5. The control of each finger is limited to triggering the grasp and there is no capacity for more granular control for the finger. Similarly, the behaviour of the lateral movement is fixed once initially triggered. Though this limits the motion of the gripper, it significantly reduces the size of the solution space and the chosen implementation proves sufficient to create an adaptive motion which mitigates the effects of interception errors.

The results presented represent a significant improvement in grasping success rates, through the use of a learned, adaptive grasping motion. This technique applied to examples from existing literature hugely increases the conditions under which a robot can

autonomously grasp a moving object. Where previously very accurate interceptions were essential [6], high performance computing infrastructures were used to perform the necessary calculations sufficiently quickly [14], and exocentric sensing infrastructure was used to track the object with sufficient accuracy [18]. These results show that adapting the grasping motion during the grasp, based on real-time tactile sensor data can reduce the demands on actuation, computation, and sensor systems, thereby allowing autonomous grasping of moving objects on robotics platforms and in environments where it was previously not possible.

The results presented here clearly demonstrate the suitability of a neural network, trained using machine learning techniques, to the task of grasping a moving object. This represents a significant improvement over previous techniques, not only in the higher grasping success rates observed during testing, and shown on Tables 6.1 & 6.2, but also since the nature of machine learning techniques make this approach suitable for more complex grasping scenarios where a heuristic approach is less viable.

Though the results and discussion presented in this chapter represent a significant step in the field of adaptive grasping motions, they also highlight several avenues for critical future research. The testing presented involved grasping an object rolling on a horizontal plane, this simplified the grasping problem to a two-dimensional one, and allowed the isolation of individual variables so that their effect could be examined. Though this has been effective to this point, a critical next step in examining the effectiveness of reactive grasping motions is to apply these strategies to a three-dimensional problem. There is also several variables which have not been addressed with the testing to date, one of particular importance is the density of the tactile sensing. All testing conducted thus far has used three tactile sensors per finger, higher density tactile sensing has been demonstrated in the prior literature and could contribute to higher performance.

7 | Conclusion

The aim of this project was to explore how adaptive grasping motions, informed by real-time tactile feedback, can improve grasp robustness when grasping moving objects. It was hypothesised that an adaptive grasping motion, based on real-time tactile sensor feedback, could mitigate the effects of errors in the interception of a ball by a robotic gripper, resulting in a more robust control strategy for grasping moving objects.

This was further broken down into a set of distinct objectives which were outlined in Chapter 1. The following section reiterates these objectives, and discusses how the research outlined in this thesis addresses each of them.

7.1 Reflection on the contributions of this thesis

Develop an experimental methodology to systematically quantify the performance of grasping strategies under a range of grasping conditions.

The thesis outlined a novel experimental methodology that aimed to systematically test and quantify the robustness of grasping control strategies for moving objects, under a range of conditions. This methodology identified three key variables which define the conditions under which grasping occurs. These are: the position of initial contact between the object and the gripper, the time which the grasping motion is initiated, and the object speed. These were independent variables for testing, enabling the robustness of each control strategy to be quantified. This systematic approach to performance evaluation is unique in the literature and has been demonstrated to offer key insights into the

performance of a range of control strategies. Furthermore, it is adaptable and well suited to the evaluation of grippers and their control strategies for grasping moving objects more broadly.

Develop a simulated robotic gripper and testing environment capable of implementing the aforementioned methodology.

A simulated gripper and test environment was developed which served several purposes including benchmarking the performance of the predictive control strategy, and evaluating the reactive control strategy. The inexpensive and automated nature of simulation allowed testing of the hypothesis using large sample sizes which would otherwise have been unfeasible. Furthermore, this simulation was later adapted to train a neural network using reinforcement learning, making the large number of training cycles required for reinforcement learning possible.

Design and manufacture a robotic gripper and testing apparatus to replicate the simulated testing in the real-world utilising the same experimental methodology.

A real-world testing apparatus, including a robotic gripper, was designed and manufactured. The gripper was a two-fingered gripper, while the testing apparatus included mechanisms to control the speed of the object, change the initial contact between object and gripper and accurately monitor the position of the ball as it approached the gripper. This testing apparatus has proven effective at addressing the questions posed in this thesis.

Explore the performance of traditional control strategies and examine their robustness to different grasping conditions

Testing was conducted, both in simulation and in the real-world, which examined the performance of a traditional, predictive grasping strategy, using the methodology outlined in Chapter 3. Its performance was evaluated at a range of test speeds, gripper-object

contact positions and grasp timings. The results obtained from both simulated and real-world testing demonstrate that high grasping success rates are possible when grasping conditions are close to optimum, i.e. low or no induced interception errors. However, the observed grasping success rates drops quickly as errors are introduced into the grasping conditions. The implications of this are that these strategies struggle to perform when conditions are not optimum, and are useful only in structured conditions.

Propose and explore strategies which leverage tactile sensor feedback to create adaptive grasping motions which react to grasping conditions in real-time.

This thesis proposes two reactive grasping strategies, one which is based on set of heuristics and a second which is based on a neural network. Real-time tactile sensing is used during the grasp to monitor the position of the ball, enable the grasping strategy to adapt the motion of the gripper appropriately. When evaluated, these strategies clearly demonstrated the ability mitigate the negative effects of non-optimum gripper-object interceptions. Furthermore, tactile sensing is proven a reliable method to gather the necessary data for such a control paradigm.

Analysis of the performance of reactive control strategies and compare their performance to that of traditional control strategies

The heuristic-based, reactive grasping strategy, was tested both in simulation and on the real-world testing apparatus. Its performance under a range of grasping conditions was evaluated. Results clearly show that the tactile sensing was able to detect interception errors in real-time, and enable the grasping strategy to mitigate their effects. This resulted in higher average grasping success rates for the heuristic-based reactive control strategy compared to its predictive counterpart. Despite this, results also showed poorer performance, compared to the predictive strategy, at specific lower spatial errors. This revealed a source of error which is inherent in a grasping strategy relying on tactile sensing to trigger the grasp. Testing revealed that the optimum time to initiate the grasp exists at some point before contact between gripper and object. A strategy which relies

on tactile sensing to trigger the grasp will always grasp after the optimum. Potential solutions to this inherent temporal error are explored in the next section.

Design and develop a control strategy based on a machine learning agent and train that agent to create an optimised adaptive grasping motion based on real-time tactile sensor feedback

Control strategies which leverage machine learning techniques were also explored. Popular in the prior art for their ability to perform effectively in complex applications and adapt to previously unseen data, machine learning techniques seem to be a good fit for this application. An artificial neural network was trained, first using synthetic data and supervised learning, then in simulation using reinforcement learning, for the task of grasping a moving object rolling toward it on a horizontal plane. It was shown that a neural network could successfully learn how to create adaptive grasping motions which are effective at mitigating the effects of interception errors. Furthermore, this shows that it is feasible to train a network to grasp a moving ball in simulation and apply it to real world gripper with minimal adaptation.

Quantify the performance of the machine learning agent compared to other control strategies.

The performance of this trained control strategy was quantified and there was an observed performance improvement when compared to the performance of both the predictive and heuristic-based reactive strategies previously tested. This demonstrated several critical findings, a neural network trained using machine learning techniques can achieve high grasping success rates in this application. This corresponds to more robust grasping despite the robot achieving a less accurate object interception. Ultimately enabling successful grasping in situations where highly accurate tracking, modeling and interceptions are impossible, i.e. in unstructured environments, for platforms with limited computational resources, or for platforms with less accurate actuation systems.

7.2 Applicability of this research

It is important when examining this research, to consider the applicability and generalisability of the contributions outlined.

The methodology developed during this research systematically examined grasping performance. It was used to examine the effect of spatial errors, temporal errors, object speed and grasping strategy on the grasping success rate of a gripper. This methodology has applicability beyond how it was used in this research. Tactile sensing implementations, gripper morphology, object shape, the target object's angle of incidence, and object spin are a few of the variables which effect grasping success rate, which haven't been examined in this research, and which could be explored using the developed methodology. A simplification of the grasping problem to a two dimensional problem is key to ensuring repeatability and isolating specific factors effecting grasping, however this simplification also limits the generalisability of the results. Though efforts where made to ensure that the conclusions drawn from testing of a two-dimensional grasping problem could be extrapolated and applied in three-dimensions, this assumption should be confirmed.

A significant advantage of this research is that it considered the task of grasping a moving object in an application agnostic way. By abstracting from the application and examining the interaction between gripper and object, the effect of specific parameters of the grasping conditions and the impact of different grasping strategies can be explored. This maximises the generalisability of these findings by exploring the fundamentals of how to autonomously grasp a moving object, dealing with challenges which are present regardless of the specific application. A drawback of conducting the research in this way is that any additional challenges which are specific to an application are not examined. This research found that adaptive grasping strategies can improve grasp robustness in the face of interception errors, and that these strategies can be enabled by real-time tactile feedback. Additional development is required to apply these findings to any particular application.

A ball was used as the target object, since a sphere presents the same grasping problem regardless of orientation. It is common in grasping research to conduct foundational research with a simple object, like a ball. The same methodology can be applied to explore more complex objects, while the research presented here using a spherical object provides a foundation from which future research can build upon and explore the effect of the size, weight, and shape of the target object.

The effectiveness of fundamental components of an adaptive grasping strategy, i.e. the lateral motion of the gripper and finger coordination, were verified in this research. A foundational understanding of their ability to mitigate the negative effects of interception errors was established. The specific implementation of these techniques used in this research was developed for a particular combination of gripper design, target object and grasping problem. A different robotic platform, tackling a different grasping problem will require a custom implementation of an adaptive grasping motion. The findings of this research are essential and can be leveraged to enable the development of these new adaptive grasping motions.

7.3 Limitations and future work

The research presented in this thesis provides novel and foundational research in adaptive grasping motions and the potential contribution of tactile sensing in the context of grasping moving objects. This section outlines the limitations of the research presented and how these should be addressed by future research.

This research simplified the grasping operation for the purposes of repeatability. It is necessary to evaluate the same techniques in more complex grasping scenarios to fully realise their utility. This would require adaption of the current implementations of the control strategies. A heuristic-based reactive strategy, similar to that proposed in this thesis, operating in a more complex grasping scenario must account for the closing motion of any additional fingers and for movement in 3D rather than the existing lateral motion. A neural network strategy would require the network to be retrained as well as the

network's input and output vectors to be adapted to account for additional tactile sensors and actuators.

Aside from the two-dimensional nature of the grasping problem, a further simplification made was the test object used, which in these experiments was a sphere. This allows the strategy to ignore the orientation of the object and focus entirely on the remaining aspects of the grasp. Though this was essential for the experiments conducted and is a commonly used technique in the prior art, in real-world applications the autonomous grasping of dynamic objects will not necessarily be limited to spherical objects. The findings and techniques demonstrated here should be incorporated into existing strategies for grasping irregular objects and their effectiveness reexamined.

The approach taken to grasping in this research involved intercepting the object, where the trajectory of the object is perpendicular to the gripper and the object collides with the static gripper. Prior research has demonstrated techniques which structure the gripper-object interaction so as to absorb the kinetic energy of the object. This increases the amount of time the object is in the graspable range of the gripper and increases the chances of a successful grasp. Tactile sensor feedback and reactive motions have the potential to contribute to structuring the interaction in this way, but the effectiveness of this approach has yet to be explored.

Though the testing conducted on reactive strategies outlined in this thesis has shown generally positive results, even the highest performing strategies (the machine learning control strategy) still under-performs compared to the traditional predictive strategy under some test conditions, i.e. in cases where there are no errors. This is due to the error in grasp timing which is inherent in any strategy relying on contact to trigger the grasp. Future research should examine a potential hybrid solution which could leverage the advantages of both strategies, by attempting to quantify the level of uncertainty in any particular scenario and act accordingly. The simplest version of this might use a predictive strategy when there is a high level of certainty associated with the interception point and the position of the gripper and resorting to a reactive strategy in more uncertain

conditions, though more complex hybrid solutions should also be explored.

A parameter which was not explored was the implementation of the tactile sensing. Results of experimental testing in both simulation and in the real-world highlight the limitations of the low density tactile sensing used in this research. The effects of tactile sensor density have yet to be explored. Furthermore, this research limited its examination to contact style tactile sensor, i.e. the sensors report contact/no-contact. Future research should examine the potential contribution of a sensor which reports a continuous reading corresponding to the normal force experienced by the tactile sensor. Similarly, the control strategies tested only considered forces in the normal direction, though shear forces in other directions might further improve the adaptive motion. Finally, proximity sensors are potentially suitable for this application. This would have the advantage of informing the grasping strategy about the real-time position of the object prior to contact, which could yield a major performance improvement.

Bibliography

- [1] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, 2019. [Online]. Available: <https://science.sciencemag.org/content/364/6446/eaat8414>
- [2] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," *Conference on Robot Learning*, vol. abs/1809.10790, 2018. [Online]. Available: <http://arxiv.org/abs/1809.10790>
- [3] Q. Lei, J. Meijer, and M. Wisse, "A survey of unknown object grasping and our fast grasping algorithm-c shape grasping," in *3rd International Conference on Control, Automation and Robotics (ICCAR)*, 2017, pp. 150–157.
- [4] Y. Yu, Z. Cao, S. Liang, W. Geng, and J. Yu, "A novel vision-based grasping method under occlusion for manipulating robotic system," *IEEE Sensors Journal*, vol. 20, no. 18, pp. 10 996–11 006, 2020.
- [5] H. Cuevas-Velasquez, N. Li, R. Tylecek, M. Saval-Calvo, and R. B. Fisher, "Hybrid multi-camera visual servoing to moving target," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1132–1137.
- [6] B. Bäuml, O. Birbach, T. Wimböck, U. Frese, A. Dietrich, and G. Hirzinger, "Catching flying balls with a mobile humanoid: System overview and design considerations," in *11th IEEE-RAS International Conference on Humanoid Robots*, 2011, pp. 513–520.

- [7] S. Kim, A. Shukla, and A. Billard, "Catching objects in flight," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1049–1065, 2014.
- [8] S.-T. Kao and M.-T. Ho, "Ball-catching system using image processing and an omni-directional wheeled mobile robot," *Sensors*, vol. 21, no. 9, 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/9/3208>
- [9] K. Mironov and M. Pongratz, "Fast knn-based prediction for the trajectory of a thrown body," in *24th Mediterranean Conference on Control and Automation (MED)*, 2016, pp. 512–517.
- [10] R. Gayanov, K. Mironov, K. Mironov, and D. Kurennov, "Estimating the trajectory of a thrown object from video signal with use of genetic programming," in *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2017, pp. 134–138.
- [11] K. Deguchi, H. Sakurai, and S. Ushida, "A goal oriented just-in-time visual servoing for ball catching robot arm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3034–3039.
- [12] C. Smith and H. I. Christensen, "Using cots to construct a high performance robot arm," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 4056–4063.
- [13] R. C. Luo and C.-H. Liao, "Robotic conveyor tracking with dynamic object fetching for industrial automation," in *IEEE 15th International Conference on Industrial Informatics (INDIN)*, 2017, pp. 369–374.
- [14] B. Bäuml, T. Wimböck, and G. Hirzinger, "Kinematically optimal catching a flying ball with a hand-arm-system," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2592–2599.
- [15] S.-T. Kao, Y. Wang, and M.-T. Ho, "Ball catching with omni-directional wheeled mobile robot and active stereo vision," in *IEEE 26th International Symposium on Industrial Electronics (ISIE)*, 2017, pp. 1073–1080.

- [16] S. Escaida Navarro, D. Weiss, D. Stogl, D. Milev, and B. Hein, "Tracking and grasping of known and unknown objects from a conveyor belt," in *ISR/Robotik 2014; 41st International Symposium on Robotics*, 2014, pp. 1–8.
- [17] J. Kober, M. Glisson, and M. Mistry, "Playing catch and juggling with a humanoid robot," in *12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, 2012, pp. 875–881.
- [18] K. Dong, K. Pereida, F. Shkurti, and A. P. Schoellig, "Catch the ball: Accurate high-speed motions for mobile manipulators via inverse dynamics learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 6718–6725.
- [19] L. Fu and J. Zhao, "Robot compliant catching by maxwell model based cartesian admittance control," *Assembly Automation*, vol. 41, no. 2, pp. 133–143, 2020.
[Online]. Available: <https://doi.org/10.1108/AA-04-2019-0062>
- [20] Y. Imai, A. Namiki, K. Hashimoto, and M. Ishikawa, "Dynamic active catching using a high-speed multifingered hand and a high-speed vision system," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 2, 2004, pp. 1849–1854 Vol.2.
- [21] K. Koyama, K. Murakami, T. Senoo, M. Shimojo, and M. Ishikawa, "High-Speed, Small-Deformation Catching of Soft Objects Based on Active Vision and Proximity Sensing," *Internal Conference on Robotics and Automation (ICRA2019)*, vol. 4, no. 2, pp. 578–585, 2019.
- [22] S. S. M. Salehian, M. Khoramshahi, and A. Billard, "A dynamical system approach for softly catching a flying object: Theory and experiment," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 462–471, 2016.
- [23] M. Riley and C. G. Atkeson, "Robot catching: Towards engaging human-humanoid interaction," *Autonomous Robots*, vol. 12, no. 1, pp. 119–128, 2002.

- [24] B. Hove and J.-J. E. Slotine, "Experiments in robotic catching," in *1991 American Control Conference*, 1991, pp. 380–386.
- [25] U. Frese, B. Bauml, S. Haidacher, G. Schreiber, I. Schaefer, M. Hahnle, and G. Hirzinger, "Off-the-shelf vision for a robotic ball catcher," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 3, 2001, pp. 1623–1629 vol.3.
- [26] Y.-B. Jia, M. Gardner, and X. Mu, "Batting an in-flight object to the target," *The International Journal of Robotics Research*, vol. 38, no. 4, pp. 451–485, 2019.
[Online]. Available: <https://doi.org/10.1177/0278364918817116>
- [27] K. Mironov and M. Pongratz, "Applying neural networks for prediction of flying objects trajectory," *Vestnik UGATU*, vol. 17, no. 6 (59), 2013.
- [28] P. Cigliano, V. Lippiello, F. Ruggiero, and B. Siciliano, "Robotic ball catching with an eye-in-hand single-camera system," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 5, pp. 1657–1671, 2015.
- [29] W. Hong and J.-J. E. Slotine, "Experiments in hand-eye coordination using active vision," in *Experimental Robotics IV*, O. Khatib and J. K. Salisbury, Eds. Springer Berlin Heidelberg, 1997, pp. 130–139.
- [30] R. Lampariello, D. Nguyen-Tuong, C. Castellini, G. Hirzinger, and J. Peters, "Trajectory planning for optimal robot catching in real-time," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 3719–3726.
- [31] A. Namiki and M. Ishikawa, "Robotic catching using a direct mapping from visual information to motor command," in *IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2, 2003, pp. 2400–2405 vol.2.
- [32] Y. Suzuki, K. Koyama, A. Ming, and M. Shimojo, "Grasping strategy for moving object using net-structure proximity sensor and vision sensor," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1403–1409.

- [33] S. Kim, E. Gribovskaya, and A. Billard, "Learning motion dynamics to catch a moving object," in *10th IEEE-RAS International Conference on Humanoid Robots*, 2010, pp. 106–111.
- [34] D. Reynaerts, "Control methods and actuation technology for whole-hand dexterous manipulation," 1995.
- [35] L. Birglen and T. Schlicht, "A statistical review of industrial robotic grippers," *Robotics and Computer-Integrated Manufacturing*, vol. 49, pp. 88–97, FEB 2018.
- [36] K. Tai, A.-R. El-Sayed, M. Shahriari, M. Biglarbegian, and S. Mahmud, "State of the art robotic grippers and applications," *Robotics*, vol. 5, no. 2, 2016. [Online]. Available: <https://www.mdpi.com/2218-6581/5/2/11>
- [37] J. Hughes, U. Culha, F. Giardina, F. Guenther, A. Rosendo, and F. Iida, "Soft Manipulators and Grippers: A Review," *Frontiers in Robotics and AI*, vol. 3, NOV 16 2016.
- [38] E. N. G. Melo, O. F. A. Sánchez, and D. A. Hurtado, "Anthropomorphic robotic hands: a review." *Ingeniería y Desarrollo*, vol. 32, no. 2, pp. 279 – 313, 2014.
- [39] G. Salvietti, "Replicating human hand synergies onto robotic hands: A review on software and hardware strategies," *Frontiers in Neurorobotics*, vol. 12, p. 27, 2018. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fnbot.2018.00027>
- [40] B. Zhang, Y. Xie, J. Zhou, K. Wang, and Z. Zhang, "State-of-the-art robotic grippers, grasping and control strategies, as well as their applications in agricultural robots: A review," *Computers and Electronics in Agriculture*, vol. 177, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169920311030>
- [41] R. Patel, R. Curtis, B. Romero, and N. Correll, "Improving grasp performance using in-hand proximity and contact sensing," in *Robotic Grasping and Manipulation*, Y. Sun and J. Falco, Eds. Springer International Publishing, 2018, pp. 146–160.

- [42] “2f-85 and 2f-140 grippers,” 2022, [Accessed 17-01-2022]. [Online]. Available: https://robotiq.com/products/2f85-140-adaptive-robot-gripper?ref=nav_product_new_button
- [43] Omron, “Omron colaborative robots,” 2022, [Accessed 21-03-2022]. [Online]. Available: <https://robotics.omron.com/collaborativerobots/gripper-onrobot>
- [44] R. Bischoff, U. Huggenberger, and E. Prassler, “Kuka youbot - a mobile manipulator for research and education,” in *IEEE International Conference on Robotics and Automation*, 2011, pp. 1–4.
- [45] “Assembly and operating manual sdh2 servo-electric 3-finger gripping hand,” 2018. [Online]. Available: <https://schunk.com/fileadmin/pim/docs/IM0023480.PDF>
- [46] “Barretthand bh8-280/282 kinematics, joint ranges, conversion factors,” May 2021, [Accessed 2021-07-06]. [Online]. Available: <https://support.barrett.com/wiki/Hand/280/KinematicsJointRangesConversionFactors>
- [47] “3-finger adaptive robot gripper,” 2022, [Accessed 17-01-2022]. [Online]. Available: https://robotiq.com/products/3-finger-adaptive-robot-gripper?ref=nav_product_new_button
- [48] “Barretthand,” 2018, [Accessed 2021-07-06]. [Online]. Available: <https://support.barrett.com/wiki/Hand>
- [49] J. Butterfass, M. Grebenstein, H. Liu, and G. Hirzinger, “Dlr-hand ii: next generation of a dextrous robot hand,” in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 1, 2001, pp. 109–114 vol.1.
- [50] “Allegro hand: Wonik robotics,” 2021, [Accessed 02-04-2022]. [Online]. Available: <https://www.wonikrobotics.com/research-robot-hand>
- [51] T. P. Tomo, W. K. Wong, A. Schmitz, H. Kristanto, A. Sarazin, L. Jamone, S. Somlor, and S. Sugano, “A modular, distributed, soft, 3-axis sensor system for

- robot hands," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 454–460.
- [52] S. R. Company, *Shadow Dexterous Hand Technical Specification*. Shadow Robot Company, 2019, [Accessed 2021-07-06]. [Online]. Available: https://www.shadowrobot.com/wp-content/uploads/shadow_dexterous_hand_technical_specification_E_20190221.pdf
- [53] A. G. Eguíluz, I. Rañó, S. A. Coleman, and T. M. McGinnity, "Reliable object handover through tactile force sensing and effort control in the shadow robot hand," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 372–377.
- [54] S. W. Ruehl, C. Parlitz, G. Heppner, A. Hermann, A. Roennau, and R. Dillmann, "Experimental evaluation of the schunk 5-finger gripping hand for grasping tasks," in *IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, 2014, pp. 2465–2470.
- [55] J. C. V. Tieck, K. Secker, J. Kaiser, A. Roennau, and R. Dillmann, "Soft-grasping with an anthropomorphic robotic hand using spiking neurons," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2894–2901, 2021.
- [56] H. Iwata and S. Sugano, "Design of human symbiotic robot twenty-one," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 580–586.
- [57] M. Maggiali, L. Fiorio, U. Pattacini, and A. Derito, *Technical Specifications of the iCub Platform*. Istituto Italiano di Tecnologia, 2019, vol. 2.5, accessed: 2021-07-06. [Online]. Available: ["https://www.iit.it/documents/175012/528824/Technical-specs_iCub_robot_Rev_2.3_05082019+%281%29.pdf/44d1e600-7a09-12e0-5eeb-5e3a2444e78c"](https://www.iit.it/documents/175012/528824/Technical-specs_iCub_robot_Rev_2.3_05082019+%281%29.pdf/44d1e600-7a09-12e0-5eeb-5e3a2444e78c)
- [58] L. Jamone, G. Metta, F. Nori, and G. Sandini, "James: A humanoid robot acting over an unstructured world," in *6th IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 143–150.

- [59] C. Stürzer, "Institution of robotics and mechatronics hand ii," 2021, [Accessed 2021-07-06]. [Online]. Available: https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-9656/16605_read-40504/
- [60] Kezi, "Allegro hand," Feb 2021, [Accessed: 2021-07-06]. [Online]. Available: <http://wiki.ros.org/Robots/AllegroHand>
- [61] I. Spectrum, "Shadow hand," May 2018, [Accessed 2021-07-06]. [Online]. Available: <https://robots.ieee.org/robots/shadow/?gallery=photo1>
- [62] "Assembly and operating manual svh servo-electric 5-finger gripping hand," 2019. [Online]. Available: <https://schunk.com/fileadmin/pim/docs/IM0020890.PDF>
- [63] "Twendy one," May 2018, [Accessed 2021-07-06]. [Online]. Available: <https://robots.ieee.org/robots/twendyone/?gallery=photo1>
- [64] "icub," May 2018, [Accessed 2021-07-06]. [Online]. Available: <https://robots.ieee.org/robots/icub/?gallery=photo1>
- [65] E. Brown, N. Rodenberg, J. Amend, A. Mozeika, E. Steltz, M. R. Zakin, H. Lipson, and H. M. Jaeger, "Universal robotic gripper based on the jamming of granular material," *Proceedings of the National Academy of Sciences*, vol. 107, no. 44, pp. 18 809–18 814, 2010. [Online]. Available: <https://www.pnas.org/doi/abs/10.1073/pnas.1003250107>
- [66] D. Grossman, "Octogripper, the octopus-inspired robotic arm, is here," Feb 2018. [Online]. Available: <https://www.popularmechanics.com/technology/gadgets/a25888/inspired-by-an-octopus-the-octogripper-is-here/>
- [67] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 161–185, MAR JAN. [Online]. Available: <https://doi.org/10.1177/0278364915592961>

- [68] H. Nakamoto, M. Ohtake, K. Komoda, A. Sugahara, and A. Ogawa, "A gripper system for robustly picking various objects placed densely by suction and pinching," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 6093–6098.
- [69] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Analysis and observations from the first amazon picking challenge," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 172–188, 2018.
- [70] C. H. Corbato, M. Bharatheesha, J. van Egmond, J. Ju, and M. Wisse, "Integrating different levels of automation: Lessons from winning the amazon robotics challenge 2016," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4916–4926, 2018.
- [71] "The vgc10 electric vacuum gripper," 2022, [Accessed 22-03-2022]. [Online]. Available: <https://onrobot.com/en/products/vgc10>
- [72] R. Tedrake, *Underactuated Robotics: Learning, Planning, and Control for Efficient and Agile Machines*. MIT, 2016, ch. Fully Actuated vs. Underactuated Systems, p. 5–6.
- [73] T. Mouri, H. Kawasaki, K. Yoshikawa, J. Takai, and S. Ito, "Anthropomorphic robot hand: Gifu hand iii," *Proc. of Int. Conf. ICCAS2002*, pp. 1288–1293, 01 2002.
- [74] H. Kawasaki, T. Komatsu, K. Uchiyama, and T. Kurimoto, "Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand ii," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC99)*, vol. 2, 1999, pp. 782–787 vol.2.
- [75] S. Funabashi, A. Schmitz, T. Sato, S. Somlor, and S. Sugano, "Robust in-hand manipulation of variously sized and shaped objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 257–263.

- [76] A. Schmitz, Y. Bansho, K. Noda, H. Iwata, T. Ogata, and S. Sugano, "Tactile object recognition using deep learning and dropout," in *IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 1044–1050.
- [77] S. "Funabashi, Y. Gang, A. Geier, A. Schmitz, and S. Sugano, "Morphology-specific convolutional neural networks for tactile object recognition with a multi-fingered hand," *International Conference on Robotics and Automation (ICRA)*, pp. 57–63, 2019.
- [78] T. P. Tomo, A. Schmitz, W. K. Wong, H. Kristanto, S. Somlor, J. Hwang, L. Jamone, and S. Sugano, "Covering a robot fingertip with uskin: A soft electronic skin with distributed 3-axis force sensitive elements for robot hands," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 124–131, 2018.
- [79] A. Schmitz, M. Maggiali, L. Natale, B. Bonino, and G. Metta, "A tactile sensor for the fingertips of the humanoid robot icub," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2212–2217.
- [80] R. Juan Antonio Corrales, M. Fernando Torres, and P. Véronique, "Finger readjustment algorithm for object manipulation based on tactile information." *International Journal of Advanced Robotic Systems*, vol. 10, 2013.
- [81] Y. Yang, "Robot physical interaction through the combination of vision, tactile and force feedback: Applications to assistive robotics," *Industrial Robot: An International Journal*, vol. 40, no. 3, Jan 2013. [Online]. Available: <https://doi.org/10.1108/ir.2013.04940caa.009>
- [82] Y. Bai and C. K. Liu, "Dexterous manipulation using both palm and fingers," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1560–1565.
- [83] R. Kōiva, M. Zenker, C. Schürmann, R. Haschke, and H. J. Ritter, "A highly sensitive 3d-shaped tactile sensor," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2013, pp. 1084–1089.

- [84] ShadowRobotics, "Developments in dextrous hands for advanced robotic applications," in *Proceedings World Automation Congress, 2004.*, vol. 15, 2004, pp. 123–128.
- [85] G. Büscher, M. Meier, G. Walck, R. Haschke, and H. J. Ritter, "Augmenting curved robot surfaces with soft tactile skin," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1514–1519.
- [86] A. Delgado, C. Jara, and F. Torres, "In-hand recognition and manipulation of elastic objects using a servo-tactile control strategy," *Robotics and Computer-Integrated Manufacturing*, vol. 48, pp. 102–112, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0736584515300995>
- [87] D. Xu, G. E. Loeb, and J. A. Fishel, "Tactile identification of objects using bayesian exploration," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 3056–3061.
- [88] E. Torres-Jara, "Obrero: a platform for sensitive manipulation," in *5th IEEE-RAS International Conference on Humanoid Robots*, 2005, pp. 327–332.
- [89] E. Torres-Jara, I. Vasilescu, and R. Coral, "A soft touch: Compliant tactile sensors for sensitive manipulation," *Computer Science and Artificial Intelligence Laboratory Technical Report*, 2006.
- [90] E. Torres-Jara and L. Natale, "Sensitive Manipulation: Manipulation Through Tactile Feedback," *International Journal of Humanoid Robotics*, vol. 15, no. 1, SI, FEB 2018, 16th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Mexican Assoc Comp Vis Robot & Neural Comp, Cancun, MEXICO, NOV 15-17, 2016.
- [91] "Motion capture systems," 2022, [Accessed 16-01-2022]. [Online]. Available: <https://optitrack.com/>
- [92] R. Platt, A. H. Fagg, and R. A. Grupen, "Null-space grasp control: Theory and experiments," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 282–295, 2010.

- [93] G. Westling and R. S. Johansson, "Factors influencing the force control during precision grip." *Experimental Brain Research*, vol. 53, no. 2, pp. 277 – 284, 1984.
- [94] J. B. van Erp and H. A. van Veen, "Touch down: The effect of artificial touch cues on orientation in microgravity," *Neuroscience Letters*, vol. 404, no. 1, pp. 78–82, 2006. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0304394006005490>
- [95] M. Bauza, O. Canal, and A. Rodriguez, "Tactile mapping and localization from high-resolution tactile imprints," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3811–3817.
- [96] P. Dallaire, P. Giguère, D. Émond, and B. Chaib-draa, "Autonomous tactile perception: A combined improved sensing and bayesian nonparametric approach," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 422–435, 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889013002285>
- [97] Y. Gao, L. A. Hendricks, K. J. Kuchenbecker, and T. Darrell, "Deep learning for tactile understanding from visual and haptic data," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 536–543, 2016.
- [98] C. Blanes, V. Cortés, C. Ortiz, M. Mellado, and P. Talens, "Non-destructive assessment of mango firmness and ripeness using a robotic gripper," *Food and Bioprocess Technology*, vol. 8, no. 9, pp. 1914–1924, Sep 2015. [Online]. Available: <https://doi.org/10.1007/s11947-015-1548-2>
- [99] C. H. Lin, T. W. Erickson, J. A. Fishel, N. Wettels, and G. E. Loeb, "Signal processing and fabrication of a biomimetic tactile sensor array with thermal, force and microvibration modalities," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2009, pp. 129–134.
- [100] H. Khamis, B. Xia, and S. J. Redmond, "Real-time friction estimation for grip force control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1608–1614.

- [101] F. Veiga, H. van Hoof, J. Peters, and T. Hermans, "Stabilizing novel objects by learning to predict tactile slip," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 5065–5072.
- [102] E. D. Engeberg and S. G. Meek, "Adaptive sliding mode control for prosthetic hands to simultaneously prevent slip and minimize deformation of grasped objects," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 1, pp. 376–385, 2013.
- [103] O. Taiyu, T. Hidetoshi, N. Kentaro, T. Tomoyuki, M. Kiyoshi, and S. Isao, "A tactile sensor using piezoresistive beams for detection of the coefficient of static friction." *Sensors (14248220)*, vol. 16, no. 5, p. 718, 2016.
- [104] P. Sikka, H. Zhang, and S. Sutphen, "Tactile servo: Control of touch-driven robot motion," in *Experimental Robotics III*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 219–233.
- [105] G. Sutanto, N. Ratliff, B. Sundaralingam, Y. Chebotar, Z. Su, A. Handa, and D. Fox, "Learning latent space dynamics for tactile servoing," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3622–3628.
- [106] N. F. Lepora, K. Aquilina, and L. Cramphorn, "Exploratory tactile servoing with active touch," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1156–1163, 2017.
- [107] H. Dang and P. K. Allen, "Stable grasping under pose uncertainty using tactile feedback," *Autonomous Robots*, vol. 36, no. 4, pp. 309–330, Apr 2014. [Online]. Available: <https://doi.org/10.1007/s10514-013-9355-y>
- [108] V. Kumar, T. Hermans, D. Fox, S. Birchfield, and J. Tremblay, "Contextual reinforcement learning of visuo-tactile multi-fingered grasping policies," 2019.
- [109] E. Kolycheva née Nikandrova and V. Kyrki, "Task-specific grasping of simiiar objects by probabiiistic fusion of vision and tactiie measurements," in *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, Nov 2015, pp. 704–710.

- [110] N. F. Lepora, A. Church, C. de Kerckhove, R. Hadsell, and J. Lloyd, "From pixels to percepts: Highly robust edge perception and contour following using deep learning and an optical biomimetic tactile sensor," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2101–2107, 2019.
- [111] Q. Li, R. Haschke, and H. Ritter, "A visuo-tactile control framework for manipulation and exploration of unknown objects," in *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 610–615.
- [112] K. Hauser, "Bayesian tactile exploration for compliant docking with uncertain shapes," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1084–1096, 2019.
- [113] M. Diftler, C. Culbert, R. Ambrose, R. Platt, and W. Bluethmann, "Evolution of the nasa/darpa robonaut control system," in *IEEE International Conference on Robotics and Automation*, vol. 2, 2003, pp. 2543–2548 vol.2.
- [114] Y. Chebotar, O. Kroemer, and J. Peters, "Learning robot tactile sensing for object manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3368–3375.
- [115] D. Cockbum, J.-P. Roberge, T.-H.-L. Le, A. Maslyczyk, and V. Duchaine, "Grasp stability assessment through unsupervised feature learning of tactile images," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2238–2244.
- [116] Y. Bekiroglu, A. Damianou, R. Detry, J. A. Stork, D. Kragic, and C. H. Ek, "Probabilistic consolidation of grasp experience," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 193–200.
- [117] Y. Bekiroglu, J. Laaksonen, J. A. Jorgensen, V. Kyrki, and D. Kragic, "Assessing grasp stability based on learning and haptic data," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 616–629, 2011.
- [118] Q. Li, O. Kroemer, Z. Su, F. F. Veiga, M. Kaboli, and H. J. Ritter, "A review of

- tactile information: Perception and action through touch," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1619–1634, 2020.
- [119] P. Roberts, M. Zadan, and C. Majidi, "Soft tactile sensing skins for robotics," *Current Robotics Reports*, vol. 2, no. 3, pp. 343–354, Sep 2021. [Online]. Available: <https://doi.org/10.1007/s43154-021-00065-2>
- [120] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing—from humans to humanoids," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 1–20, 2010.
- [121] Z. Kappassov, J.-A. Corrales, and V. Perdereau, "Tactile sensing in dexterous robot hands — review," *Robotics and Autonomous Systems*, vol. 74, pp. 195–220, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889015001621>
- [122] A. Yamaguchi and C. G. Atkeson, "Recent progress in tactile sensing and sensors for robotic manipulation: can we turn tactile sensing into vision?" *Advanced Robotics*, vol. 33, no. 14, pp. 661 – 673, 2019.
- [123] T. Paulino, P. Ribeiro, M. Neto, S. Cardoso, A. Schmitz, J. Santos-Victor, A. Bernardino, and L. Jamone, "Low-cost 3-axis soft tactile sensors for the human-friendly robot vizzy," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 966–971.
- [124] "Mlx90393 micropower triaxis magnetometer datasheet," 2017, [Accessed 02-04-2022]. [Online]. Available: <https://cdn.sparkfun.com/assets/e/e/f/2/7/MLX90393-Datasheet-Melexis.PDF>
- [125] K. Takahashi and J. Tan, "Deep visuo-tactile learning: Estimation of tactile properties from images," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8951–8957.
- [126] S. Funabashi, S. Ogasa, T. Isobe, T. Ogata, A. Schmitz, T. P. Tomo, and S. Sugano, "Variable in-hand manipulations for tactile-driven robot hand via

cnn-lstm,” in *IEEE International Conference on Intelligent Robots and Systems (IROS 2020)*, 2020, pp. 9472–9479.

- [127] “Uskin curved,” 2022, [Accessed 23-03-2022]. [Online]. Available: <https://xelarobotics.com/en/uskin-curved>
- [128] T. Ishihara, A. Namiki, M. Ishikawa, and M. Shimojo, “Dynamic pen spinning using a high-speed multifingered hand with high-speed tactile sensor,” in *6th IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 258–263.
- [129] J. Youngdo, L. Duck-Gyu, P. Jonghwa, K. Hyunhyub, and L. Hyuneui, “Piezoresistive tactile sensor discriminating multidirectional forces.” *Sensors* (14248220), vol. 15, no. 10, pp. 25 463 – 25 473, 2015.
- [130] R. S. Dahiya, D. Cattin, A. Adami, C. Collini, L. Barboni, M. Valle, L. Lorenzelli, R. Oboe, G. Metta, and F. Brunetti, “Towards tactile sensing system on chip for robotic applications,” *IEEE Sensors Journal*, vol. 11, no. 12, pp. 3216–3226, 2011.
- [131] G. Canavese, S. Stassi, C. Fallauto, S. Corbellini, V. Cauda, V. Camarchia, M. Pirola, and C. F. Pirri, “Piezoresistive flexible composite for robotic tactile applications,” *Sensors and Actuators A: Physical*, vol. 208, pp. 1–9, 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0924424713005724>
- [132] “Pr2,” May 2018, [Accessed 12-07-2021]. [Online]. Available: <https://robots.ieee.org/robots/pr2/>
- [133] “Pressure sensor pad: Tactarray,” 2020, [Accessed 09-07-2021]. [Online]. Available: <https://pressureprofile.com/sensors/tactarray>
- [134] “Ad7147,” 2021, [Accessed 09-07-2021]. [Online]. Available: <https://www.analog.com/en/products/ad7147.html>
- [135] “Takktile sensors,” 2020, [Accessed 12-07-2021]. [Online]. Available: <https://www.labs.righthandrobotics.com/takktile-sensors>

- [136] "Reflex robotic gripper," 2020, [Accessed 12/07/2021]. [Online]. Available: <https://www.labs.righthandrobotics.com/reflexhand>
- [137] "Takktile sensors," 2022, [Accessed 23-03-2022]. [Online]. Available: <https://softroboticstoolkit.com/book/takktile-sensors>
- [138] J.-S. Heo, J.-H. Chung, and J.-J. Lee, "Tactile sensor arrays using fiber bragg grating sensors," *Sensors and Actuators A: Physical*, vol. 126, no. 2, pp. 312–327, 2006. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0924424705006084>
- [139] E. M. Reimer and L. Danisch, "Pressure sensor based on illumination of a deformable integrating cavity," Jun 1999, [Accessed 02-04-2022]. [Online]. Available: <https://patents.google.com/patent/US5917180A/en>
- [140] Y. Ohmura, Y. Kuniyoshi, and A. Nagakubo, "Conformable and scalable tactile sensor skin for curved surfaces," in *Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 1348–1353.
- [141] "Omd-20-se-40n datasheet," 2018, [Accessed 12-07-2021]. [Online]. Available: https://www.g4.com.tw/userfiles/files/Datasheet/onrobot_3d_force_sensor_omd_20_se_40n.pdf
- [142] M. Kaboli, K. Yao, and G. Cheng, "Tactile-based manipulation of deformable objects with dynamic center of mass," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 752–757.
- [143] G. Palli, C. Melchiorri, G. Vassura, U. Scarcia, L. Moriello, G. Berselli, A. Cavallo, G. Maria, C. Natale, S. Pirozzi, C. May, F. Ficuciello, and B. Siciliano, "The dexmart hand: Mechatronic design and experimental evaluation of synergy-based control for human-like grasping." *International Journal of Robotics Research*, 2014.
- [144] A. Yamaguchi and C. G. Atkeson, "Combining finger vision and optical tactile sensing: Reducing and handling errors while cutting vegetables," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 1045–1051.

- [145] M. K. Johnson and E. H. Adelson, "Retrographic sensing for the measurement of surface texture and shape," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1070–1077.
- [146] N. F. Lepora and B. Ward-Cherrier, "Superresolution with an optical tactile sensor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 2686–2691.
- [147] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, "More than a feeling: Learning to grasp and regrasp using vision and touch," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, 2018.
- [148] R. Li, R. Platt, W. Yuan, A. ten Pas, N. Roscup, M. A. Srinivasan, and E. Adelson, "Localization and manipulation of small parts using gelsight tactile sensing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3988–3993.
- [149] R. Masuda, "Multifunctional optical proximity sensor using phase modulation," *Journal of Robotic Systems*, vol. 3, no. 2, pp. 137–147, 1986. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.4620030203>
- [150] K. Koyama, M. Shimojo, T. Senoo, and M. Ishikawa, "High-speed high-precision proximity sensor for detection of tilt, distance, and contact," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3224–3231, 2018.
- [151] P. Hebert, N. Hudson, J. Ma, T. Howard, T. Fuchs, M. Bajracharya, and J. Burdick, "Combined shape, appearance and silhouette for simultaneous manipulator and object tracking," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 2405–2412.
- [152] A. Morales, M. Prats, P. Sanz, and A. P. Pobil, "An experiment in the use of manipulation primitives and tactile perception for reactive grasping," in *International Conference on Robotics and Automation*. IEEE, 2007.

- [153] Y. Bekiroglu, J. Laaksonen, J. A. Jorgensen, V. Kyrki, and D. Kragic, "Assessing grasp stability based on learning and haptic data," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 616–629, 2011.
- [154] M. Bühler, D. E. Koditschek, and P. J. Kindlmann, "Planning and control of robotic juggling tasks," in *The fifth international symposium on Robotics research*. Cambridge, MA, USA: MIT Press, 1990, pp. 321–332.
- [155] A. Namiki and N. Itoi, "Ball catching in kendama game by estimating grasp conditions based on a high-speed vision system and tactile sensors," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 634–639.
- [156] H. Hagra, R. Ramadan, M. Nawito, H. Gabr, M. Zaher, and H. Fahmy, "A fuzzy based hierarchical coordination and control system for a robotic agent team in the robot hockey competition," in *International Conference on Fuzzy Systems*, 2010, pp. 1–8.
- [157] O. Koç, G. Maeda, and J. Peters, "Online optimal trajectory generation for robot table tennis," *Robotics and Autonomous Systems*, vol. 105, pp. 121–137, 2018.
[Online]. Available:
<https://www.sciencedirect.com/science/article/pii/S0921889017306164>
- [158] J. Kober, K. Mülling, O. Krömer, C. H. Lampert, B. Schölkopf, and J. Peters, "Movement templates for learning of hitting and batting," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 853–858.
- [159] T. Senoo, A. Namiki, and M. Ishikawa, "Ball control in high-speed batting motion using hybrid trajectory generator," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 1762–1767.
- [160] S. Schaal and C. G. Atkeson, "One-handed juggling: A dynamical approach to a rhythmic movement task," *Journal of Motor Behavior*, vol. 28, no. 2, p. 165, 1996.
- [161] R. Andersson, "Understanding and applying a robot ping-pong player's expert

controller,” in *Proceedings, 1989 International Conference on Robotics and Automation*, 1989, pp. 1284–1289 vol.3.

- [162] K. Zhang, Z. Cao, J. Liu, Z. Fang, and M. Tan, “Real-time visual measurement with opponent hitting behavior for table tennis robot,” *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 4, pp. 811–820, 2018.
- [163] B. Bishop and M. Spong, “Vision-based control of an air hockey playing robot,” *IEEE Control Systems Magazine*, vol. 19, no. 3, pp. 23–32, 1999.
- [164] D. Carneiro, F. Silva, and P. Georgieva, “The role of early anticipations for human-robot ball catching,” in *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2018, pp. 10–16.
- [165] A. A. Bhole, J. Kumle, S. S. Groothuis, and R. Carloni, “Control of a variable stiffness joint for catching a moving object,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4756–4761.
- [166] M. M. Schill and M. Buss, “Robust ballistic catching: A hybrid system stabilization problem,” *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1502–1517, 2018.
- [167] N. Ucheyama, S. Sano, and K. Ryuman, “Adaptive control of a robotic manipulator for soft catching of a falling object,” *Mechanical Engineering Journal*, vol. 2, no. 3, 2015. [Online]. Available: <https://doi.org/10.1299/mej.14-00074>
- [168] L. Mlodinow and I. Stamatescu, “An evolutionary procedure for machine learning,” *International Journal of Computer and Information Sciences*, vol. 14, no. 4, pp. 201–219, 1985. [Online]. Available: <https://doi.org/10.1007/BF00997019>
- [169] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. S. Awwal, and V. K. Asari, “A state-of-the-art survey on deep learning theory and architectures,” *Electronics*, vol. 8, no. 3, 2019. [Online]. Available: <https://www.mdpi.com/2079-9292/8/3/292>

- [170] Y. Li, Q. Lei, C. Cheng, G. Zhang, W. Wang, and Z. Xu, "A review: machine learning on robotic grasping," in *Eleventh International Conference on Machine Vision (ICMV 2018)*, vol. 11041, International Society for Optics and Photonics. SPIE, 2018, pp. 775 – 783. [Online]. Available: <https://doi.org/10.1117/12.2522945>
- [171] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, "A survey of deep learning applications to autonomous vehicle control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 712–733, 2021.
- [172] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3826–3839, 2020.
- [173] T. G. Thuruthel, B. Shih, C. Laschi, and M. T. Tolley, "Soft robot perception using embedded soft sensors and recurrent neural networks," *Science Robotics*, vol. 4, no. 26, JAN 16 2019.
- [174] A. Dziedzickis, A. Kaklauskas, and V. Bucinskas, "Human emotion recognition: Review of sensors and methods," *Sensors*, vol. 20, no. 3, p. 592, Jan. 2020. [Online]. Available: <https://doi.org/10.3390/s20030592>
- [175] S. Wan and S. Goudos, "Faster r-cnn for multi-class fruit detection using a robotic vision system," *Computer Networks*, vol. 168, p. 107036, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389128619306978>
- [176] K. Muelling, J. Kober, and J. Peters, "Learning table tennis with a mixture of motor primitives," in *10th IEEE-RAS International Conference on Humanoid Robots*, 2010, pp. 411–416.
- [177] C. Li, C. Guo, W. Ren, R. Cong, J. Hou, S. Kwong, and D. Tao, "An underwater image enhancement benchmark dataset and beyond," *IEEE Transactions on Image Processing*, vol. 29, pp. 4376–4389, 2020.

- [178] A. Shukla and A. Billard, "Coupled dynamical system based arm-hand grasping model for learning fast adaptation strategies," *Robotics and Autonomous Systems*, vol. 60, no. 3, SI, pp. 424–440, MAR 2012.
- [179] H. van Hoof, T. Hermans, G. Neumann, and J. Peters, "Learning robot in-hand manipulation with tactile features," in *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 121–127.
- [180] D. Seita, F. T. Pokorny, J. Mahler, D. Kragic, M. Franklin, J. Canny, and K. Goldberg, "Large-scale supervised learning of the grasp robustness of surface patch pairs," in *IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN)*, 2016, pp. 216–223.
- [181] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, p. 84–90, May 2017. [Online]. Available: <https://doi.org/10.1145/3065386>
- [182] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3406–3413.
- [183] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [184] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [185] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.
- [186] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

- [187] Y. LeCun, C. Cortes, and C. Burges, "The mnist database," 1998, [Accessed 02-04-2022]. [Online]. Available: <http://yann.lecun.com/exdb/mnist/>
- [188] S. Joshi, S. Kumra, and F. Sahin, "Robotic grasping using deep reinforcement learning," in *IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 2020, pp. 1461–1466.
- [189] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- [190] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018. [Online]. Available: <https://doi.org/10.1177/0278364917710318>
- [191] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
- [192] "Dex-net dataset," 2017, [Accessed 02-04-2022]. [Online]. Available: <https://berkeley.app.box.com/s/6mnb2bzi5zfa7qpwyn7uq5atb7vzbztng>
- [193] X. Zhou, X. Lan, H. Zhang, Z. Tian, Y. Zhang, and N. Zheng, "Fully convolutional grasp detection network with oriented anchor box," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7223–7230.
- [194] D. Guo, F. Sun, B. Fang, C. Yang, and N. Xi, "Robotic grasping using visual and tactile sensing," *Information Sciences*, vol. 417, pp. 274–286, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S002002551730837X>
- [195] F. E. Viña B., Y. Karayiannidis, C. Smith, and D. Kragic, "Adaptive control for pivoting with visual and tactile feedback," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 399–406.

- [196] K. Kleeberger, R. Bormann, W. Kraus, and M. F. Huber, "A survey on learning-based robotic grasping," *Current Robotics Reports*, vol. 1, no. 4, pp. 239–249, Dec 2020. [Online]. Available: <https://doi.org/10.1007/s43154-020-00021-6>
- [197] J. Borrego, R. Figueiredo, A. Dehban, P. Moreno, A. Bernardino, and J. Santos-Victor, "A generic visual perception domain randomisation framework for gazebo," in *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2018, pp. 237–242.
- [198] F. Zhang, J. Leitner, M. Milford, B. Upcroft, and P. I. Corke, "Towards vision-based deep reinforcement learning for robotic motion control," *Computing Research Repository (CoRR)*, vol. abs/1511.03791, 2015. [Online]. Available: <http://arxiv.org/abs/1511.03791>
- [199] W. C. Agboh and M. R. Dogar, "Real-time online re-planning for grasping under clutter and uncertainty," in *IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, 2018, pp. 1–8.
- [200] L. Ellekilde and H. G. Petersen, "Design and test of object aligning grippers for industrial applications," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 5165–5170.
- [201] J. A. Joergensen, L. P. Ellekilde, and H. G. Petersen, "Robworksim - an open simulator for sensor based grasping," in *41st International Symposium on Robotics (ISR) and 6th German Conference on Robotics (ROBOTIK)*, 2010, pp. 1–8.
- [202] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.
- [203] E. Rohmer, S. P. N. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1321–1326.

- [204] S. James, M. Freese, and A. J. Davison, "Pyrep: Bringing v-rep to deep robot learning," 2019, arXiv preprint arXiv:1906.11176.
- [205] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.
- [206] A. Miller and P. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.
- [207] "Webots," 2021, [Accessed 01-08-2021]. [Online]. Available: <https://cyberbotics.com/>
- [208] "Sdformat specification," 2021, [Accessed 05-08-2021]. [Online]. Available: <http://sdformat.org/spec>
- [209] B. Massa, S. Roccella, M. Carrozza, and P. Dario, "Design and development of an underactuated prosthetic hand," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 4, 2002, pp. 3374–3379 vol.4.
- [210] M. Carrozza, P. Dario, F. Vecchi, S. Roccella, M. Zecca, and F. Sebastiani, "The cyberhand: on the design of a cybernetic prosthetic hand intended to be interfaced to the peripheral nervous system," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (Cat. No.03CH37453)*, vol. 3, 2003, pp. 2642–2647 vol.3.
- [211] G. Stellin, G. Cappiello, S. Roccella, M. Carrozza, P. Dario, G. Metta, G. Sandini, and F. Becchi, "Preliminary design of an anthropomorphic dexterous hand for a 2-years-old humanoid: towards cognition," in *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, (BioRob)*, 2006, pp. 290–295.
- [212] M. Ciocarlie and P. Allen, "A constrained optimization framework for compliant underactuated grasping," *Mechanical Sciences*, vol. 2, no. 1, p. 17–26, 2011.

- [213] "SV01A103AEA01R00", 2021, [Accessed 27-07-2021]. [Online]. Available: <https://www.murata.com/en-us/api/pdfdownloadapi?cate=&partno=SV01A103AEA01R00>
- [214] C. Ledermann, S. Wirges, D. Oertel, M. Mende, and H. Woern, "Tactile sensor on a magnetic basis using novel 3d hall sensor - first prototypes and results," in *IEEE 17th International Conference on Intelligent Engineering Systems (INES)*, 2013, pp. 55–60.
- [215] L. Jamone, L. Natale, G. Metta, and G. Sandini, "Highly sensitive soft tactile sensors for an anthropomorphic robotic hand," *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4226–4233, 2015.
- [216] "Polycraft t15 translucent rtv addition cure silicone rubber shore a15," <https://www.mbf.co.uk/polycraft-t-15.html>, 2021, [Accessed: 2019-05-27].
- [217] J. Clark and D. Amodei, "Faulty reward functions in the wild," Mar 2016, [Accessed 02-04-2022]. [Online]. Available: <https://openai.com/blog/faulty-reward-functions/>

A1 | Appendix

A1.1 Manufacture of custom tactile sensors

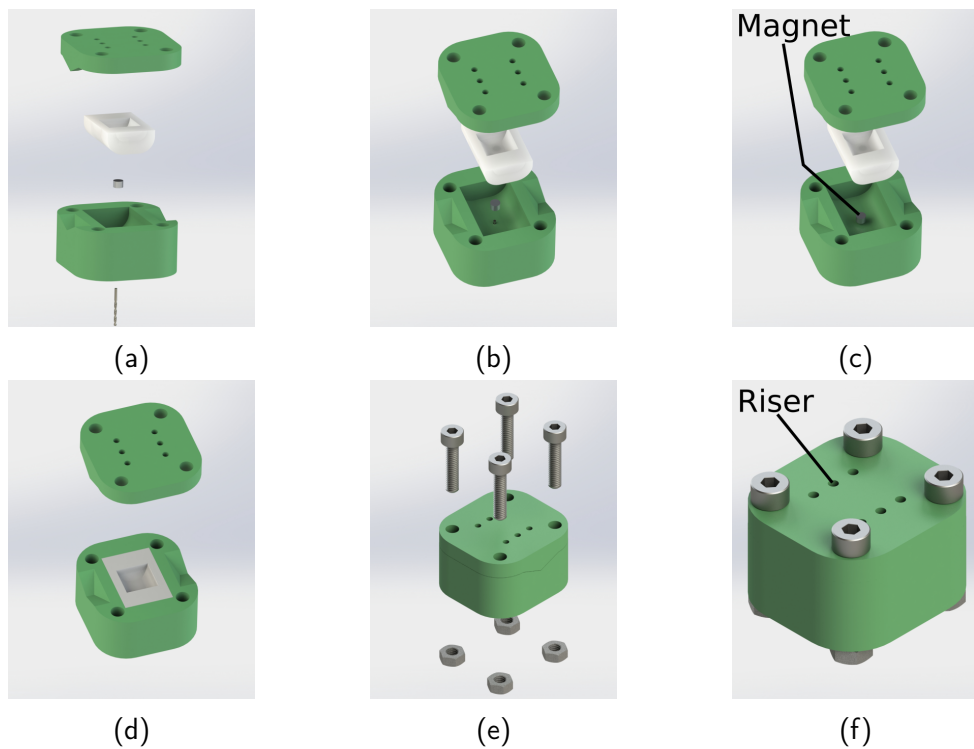


Figure A1.1: 3D Rendering of the molding process, showing (a) an exploded view (b) the drillbit in place (c) the magnet in place (d) the silicon in place with the lid off (e) the mold closed, and finally (f) the mold closed and secured with bolts

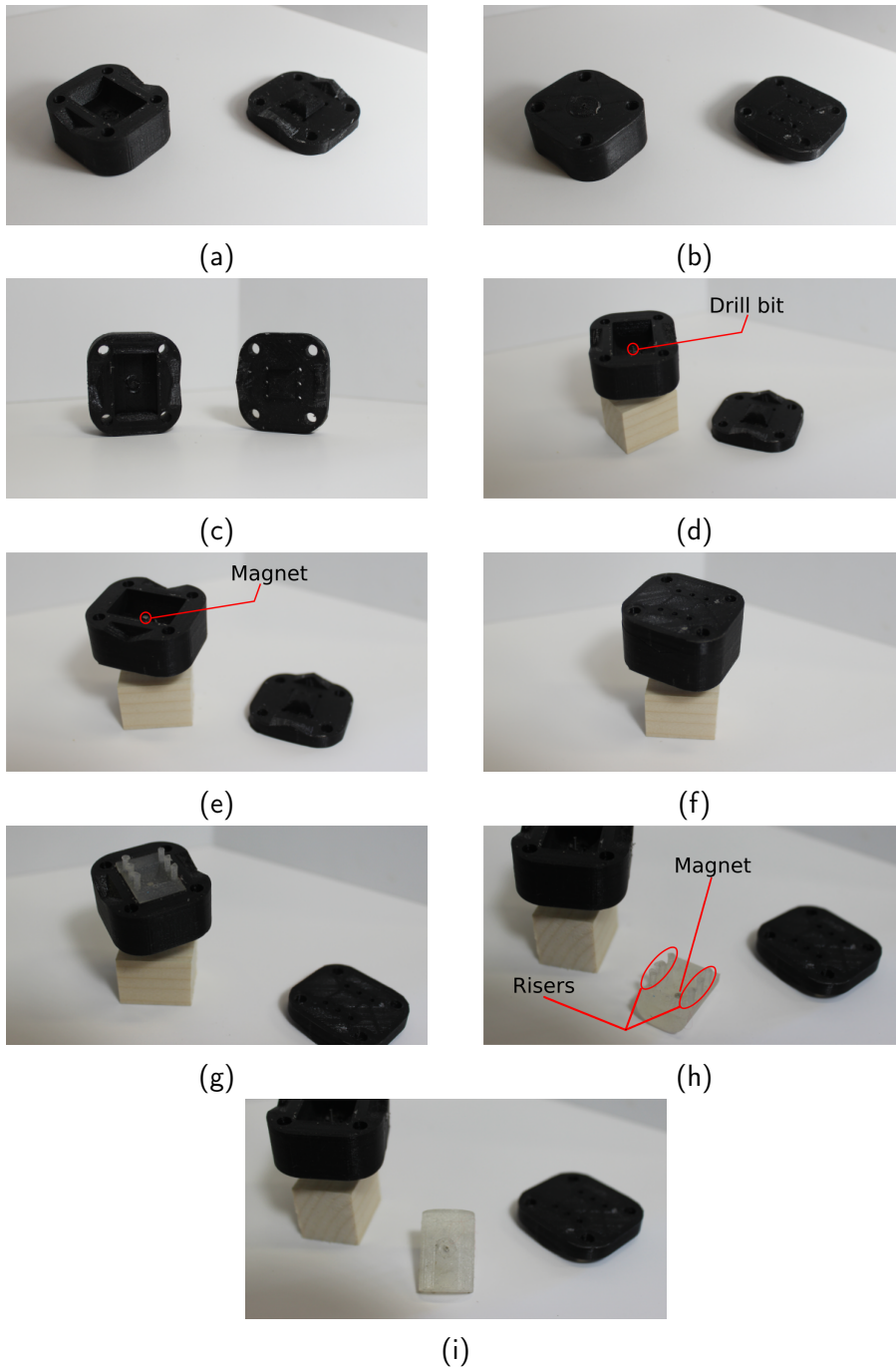


Figure A1.2: Images of the 3D printed mold and molding process. (a) mold view 1 (b) mold view 2 (c) mold view 3 (d) mold with drillbit in place (e) mold with magnet in place (f) closed mold (g) resulting silicon (h) molded silicon with labels (i) final molded silicon