



In-Hand Pose Estimation Through Purely Tactile Perception and In-Hand Manipulation

A Master Thesis

written by

Victor Melbye Staven
vista17@student.sdu.dk

The code for this project is available at
https://github.com/vmstavens/in_hand_pose_estimation

University of Southern Denmark
The Technical Faculty

Word Count : 962
May 15, 2023

Abstract

Some abstract text explaining the goal, methods and conclusion of the project.

Contents

Acknowledgments	iii
Acronyms and Terms	iv
1 Introduction	1
1.1 Context	1
1.2 Problem Description	1
1.3 Thesis Overview	3
2 System Setup	5
2.1 Simulation Setup	5
2.2 Software Setup	6
3 State of the Art	8
3.1 Problem 1 - Tactile Perception	8
3.2 Problem 2 - Pose Estimation	10
3.3 Problem 3 - In-Hand Manipulation	12
4 Modeling	13
5 Tactile Perception	17
5.1 Methods	17
5.1.1 Recursive Least Squares	17
5.1.2 Network Architecture	18
5.1.3 Network Training Procedure	20
5.2 Experimental Setup	21
5.2.1 Contact Normal Estimation	21
5.2.2 Skew Force Estimation	21
5.3 Results	22
5.3.1 Contact Normals	22
5.3.2 Skew Forces	23
5.3.3 Physics Engine Comparison	23
5.4 Discussion & Conclusion	26
6 Pose Estimation	28
6.1 Introduction	28
6.2 Method	28
6.2.1 Graduated Non-Convexity	29
6.2.2 Relaxed Convex Quadratic Programming	29
6.3 Experimental Setup	29
7 In-Hand Manipulation	30
7.1 Introduction	30
7.2 Related Work	30

8 System Integration	31
8.1 Introduction	31
9 Discussion	32
10 Conclusion	33
A Shadow Dexterous Hand - Technical Specifications	43
B Tactile Perception - Simulated Electrode Activations	45

Acknowledgements

I would like to express my sincere gratitude to those who have supported and guided me throughout my thesis project.

First and foremost, I would like to extend my deepest appreciation to my thesis supervisor, Christoffer Sloth, Lektor at SDU Robotics, for his exceptional guidance and support throughout the entire process. His expertise and knowledge in the field of robotics have been invaluable in shaping this project.

I would also like to thank Yitaek Kim for his invaluable support and sparring in solving key aspects of this project. His insights and advice have been critical to the successful completion of this thesis.

Furthermore, I would like to extend my thanks to Shadow Robotics for providing the underlying software for the project to build upon along with technical support, enabling this project. Their contribution has been essential in the successful completion of this project.

Finally, I would like to thank my family and friends for their unwavering support and encouragement throughout this journey. Their love and support have been a constant source of motivation and inspiration.

Once again, I express my deepest gratitude to all those who have played a significant role in this project.

Acronyms

AEBM analytical elasticity-based models.	MLP Multi Layered Perceptron.
CAD Computer Aided Design.	OMPL The Open Motion Planning Library.
cGAN conditional Generative Adversarial Network.	PC point cloud.
cobots collaborative robots.	PCR point cloud registration.
CP correspondence problem.	PD Proportional-Derivative.
CPD Coherent Point Drift.	PE pose estimation.
CSGM Cross-Source Graph Matching.	PID Proportional Integral Derivative.
CV computer vision.	PNP pick-and-place.
DeepGMR Deep Gaussian Mixture Registration.	PRM Probabilistic Roadmap Method.
DL deep learning.	PwoF point-contact-without-friction.
DOF degrees of freedom.	QAP Quadratic Assignment Problem.
EE end effector.	QCQP Quadratically Constrained Quadratic Programming.
EFM elastic foundation models.	QP Quadratic Programming.
FEM finite element models.	RANSAC Random Sample Consensus.
FGM Factorized Graph Matching.	RCQP Relaxed Convex Quadratic Programming.
FMT Fast Marching Tree.	ReLU Rectified Linear Unit.
FPFH Fast Point Feature Histograms.	RLS Recursive Least Squares.
GK grasp kinematics.	ROM Range of Motion.
GMM Gaussian Mixture Model.	ROS Robot Operating System.
GNC Graduated Non-Convexity.	RRT Rapidly-exploring Random Tree.
GT Ground Truth.	SDP Semi-Definite Programming.
HF hard finger.	SF soft finger.
ICP Iterative Closest Point.	SOCOP Second Order Cone Programming.
IEP Inverse Elasticity Problem.	TCP Tool Center Point.
IHM In-Hand Manipulation.	TP Tactile Perception.
JRMPC Joint Registration of Multiple Point Sets.	UR Universal Robots.
MIM Matrix Inversion Method.	
ML machine learning.	

Terms

collaborative robots (cobots) are robots which facilitate human-robot collaboration [1].

computer vision (CV) is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs - and take actions or make recommendations based on that information [2].

correspondence problem (CP) is the problem where one aims at finding correspondences between the pixels in two (or more) images [3].

deep learning (DL) are methods that allow computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction [4].

end effector (EE) is a generic term for all functional units involved in direct interaction of the robot system with the environment or with a given object [5].

manipulator : A serial robot mechanisms. The robot manipulator is represented by a serial chain of rigid bodies, called robot segments, connected by joints [6].

point cloud registration (PCR) is a generic term for all functional units involved in direct interaction of the robot system with the environment or with a given object [5].

pose estimation (PE) A particular instance of feature-based alignment, which occurs very often, is estimating an object's 3D pose from a set of 2D point projections. This pose estimation problem is also known as extrinsic calibration [7].

Robot Operating System (ROS) is a set of open-source software libraries and tools that help build robot applications. [8].

Chapter 1

Introduction

1.1 Context

As of 2022 most of the industrialized world has developed tools for unprecedented growth in wealth and technology on a global scale [9, Chapter 4]. In such times a great deal of consumerism and interconnection is present with people needing products produced faster and more consistently than ever before [9, Chapter 4]. As one would expect, this creates a high demand for manufacturers to reliably and consistently provide products, while also remaining flexible as the demand for different product change rapidly. To accommodate the need for ever-greater volumes of products, consistent, reliable and flexible labor is essential in assembly, transport and manipulation processes in the production pipeline. Due to these types of manual labor being largely done by unskilled workers, automation alternatives are being adopted which provide benefits [9, Chapter 4]. This different approach to manufacturing has been labeled the fourth industrial revolution or i4.0 for short. The beneficiaries are the employer and employee, with the employer having the benefits: Avoid paying monthly salaries to unskilled laborers doing manual tasks, here the automation alternative only requires electrical energy and potential supervision by a few qualified individuals. Potential risks are also involved when hiring humans as the workforce can be inconsistent due to human error [10] or left out due to illness etc. Considerations about workers' rights such as working conditions and wages also need not be considered. Workers furthermore cause production limitations in the form of stand-still hours, such as bathroom and lunch breaks along with after-work hours and holidays. This replacement of manual labor also potentially benefits the employee, as boring and physically wearing work is automated, enabling the employees to take on different and less wearing and potentially dangerous roles. While the issue of labor unemployment becomes apparent solutions that provide support to already hired workers have been developed, such as collaborative robots (cobots) [11] which would negate this problem.

When implementing automation of production lines using robotics, certain categories of problems are revealed. These include assembly, alteration and pick-and-place (PNP), the last being the one of interest in this project.

1.2 Problem Description

Pick-and-place manipulators are used in a wide variety of different fields such as sorting of waste [12] handling of food [13, 14] and factory bin picking [15, 16, 17]. The solutions in these industries are examples of subcategories under the PNP problem, namely sorting and bin picking. Since both of these are subcategories of the PNP problem, they fundamentally follow the same sequential four phases from start to end. These phases are pre-grasping, grasping, transport, and placement [16] for traditional implementations of the PNP pipeline. The pre-grasp phase involves localizing the object(s), potentially estimating their pose and executing the trajectory to move the end effector (EE) grasp, collision-free to said object(s). Here different potential grasps can be considered to determine the best pose for the EE. In the grasping phase, the EE grasps the object in such a manner that the object's entire weight is supported by the EE, and ends when the object no longer is in contact with the environment, which often is the container holding the object. The transportation phase involves the motion of the manipulator to move from the pose achieved after the grasping phase, to a pose ready for placement of the object in the desired placing area or fixture. Here considerations may be needed about how much force and torque the EE's grasp can tolerate while moving without losing the object. Finally, the goal of the placing phase is to place the object within the placing

area or fixture in a desired end pose. Here the constraints on the end pose might differ significantly based on the application, as the pose of greens in a crate might need less precision than if the manipulator hands a bolt to another robotics system in the pipeline.

While these phases make up a traditional PNP system, certain assumptions are made regarding the objects of interest for this pipeline to function. Specifically, the localization and pose estimation (PE) of the pre-grasp phase are assumed possible due to either ensured object poses or estimated poses through computer vision (CV) sensory system. Due to CV being a mature research field a wide range of solution proposals to these problems have been generated [18]. These include classic vision [19, 20], deep learning (DL) based [21] and combinations of these [22]. However, while these may be sufficient for certain tasks they fundamentally suffer from the weaknesses introduced by vision techniques. These are a great number of outliers caused by: occlusions, reflecting, transparent or homogeneous surfaces, and repetitive structures when solving the correspondence problem (CP). Within factory settings, the common ones are transparent and reflective objects, due to metallic, plastic and glass products often being the materials used. While DL solutions have been developed for both reflective [23] and transparent [24] objects, these are use case specific and show limited results in a wider range of applications.

This project suggests a different PNP pipeline for cases where the object's starting pose is unknown. In this PNP pipeline the PE is moved from the pre-grasping phase to a new phase between the grasping and transportation phase, called the PE phase. The specific goal of this project is to develop a solution to this phase using tactile sensors in the EE to determine the object's pose. By using tactile sensing rather than visual the problems presented above will be eliminated. This will be done using a humanoid gripper as the EE with tactile sensors in each finger, more specifically a Shadow Dexterous Hand [25] with 24 degrees of freedom (DOF).

The alternative pipeline this project will enable can be seen in the upper row of Fig. 1.1 compared to the traditional pipeline in the lower row.

In Fig. 1.1(a) the pre-grasping phase can be seen for both pipelines. Here the traditional pipeline in cases of multiple objects often will employ custom fingertips or grippers entirely to facilitate form closure grasps, due to the grippers not having the flexibility to perform reliable force closure. On the contrary force closure can be performed with a humanoid gripper on a wide range of objects with no need for changing gripper equipment.

In Fig. 1.1(b) the grasping phase can be seen which introduces a greater level of complexity when using the suggested pipeline due to the humanoid gripper being a more complex physical system to represent and control. This is compared to the simplicity of executing potential binary commands in the traditional pipeline e.g. open and close.

In Fig. 1.1(c) the transportation phase can be seen, which introduces one of the benefits of using the suggested pipeline. Here tactile sensors in a humanoid gripper can pose and estimate the object and manipulations can be performed to change the object's pose such that easier placement can be performed in the following phase. This form of object manipulation is not feasible for the simple pneumatic grippers used in a traditional pipeline.

In Fig. 1.1(d) the placement phase can be seen, which demonstrates the result of the previous phase, as the traditional pipeline now has to change the grip on the object to properly place it in the socket, while the humanoid gripper simply can insert the part, as it is already oriented properly.

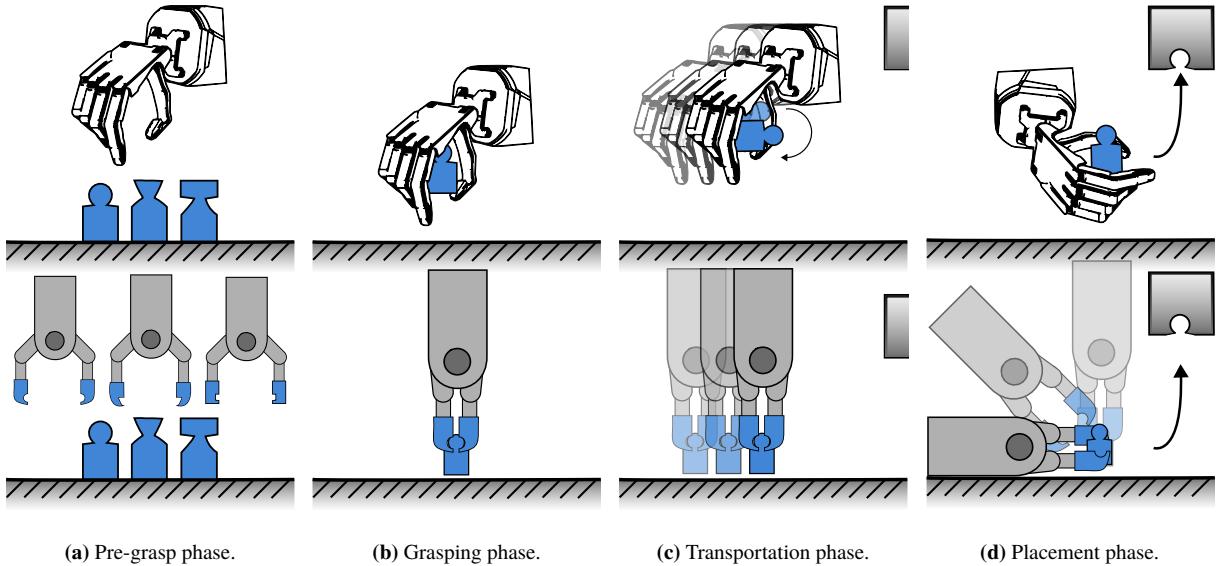


Fig. 1.1: A comparison between the traditional and suggested PNP pipeline.

To solve this PE problem, three sub-problems are identified and labeled problems 1, 2 and 3.

Problem 1 involves modeling the contact between the gripper's tactile sensors and the object, also referred to as tactile perception.

Problem 2 is to convert the collected data from problem 1 to estimated pose candidates.

Problem 3 involves in-hand manipulation. Since the initial grasp of the object might not be oriented in a manner where the recognizable features make context with the tactile sensors, manipulating the object within the EE's grasp will enable further information gathering. Thus the final problem is to control the EE in such a manner that the tactile sensors make context with the object in intelligently decided areas for a better pose estimate.

To test if the developed system successfully solves the PE problem, it is hypothesized that the intelligent probing method provides a statistically significant faster average PE convergence, along with a statistically significant greater success rate when determining the correct pose. A correct pose is here defined as the pose being greater than or equal to 95 % of the ground truth pose, and statistically significant is defined by an α -level of 95 %. This hypothesis will be referred to as H_1 , while the null hypothesis H_0 being that there is no statistically significant difference between intelligent and random probing's PE performance as described above.

1.3 Thesis Overview

This project is composed of 9 chapters, each listed below with an associated description.

Chapter 1: The introduction presents the historical context of the project along with a problem description and thesis overview. The problem description contains the decomposition of the project into sub-problems which will be addressed in the following chapters.

Chapter 2: The system setup presents an overview of the practical details of the project in the form of the system setup along with the code developed to execute the project.

Chapter 3: The literature review addresses the three problems identified in Chapter 1 by providing a thorough literature review on older as well as newer methods for solving each of the identified problems. To end each section, groups are compared and a method is chosen among the ones presented.

Chapter 4: The modeling chapter presents the physical modeling of the system and defines important mathematical notations and quantities as used in the following chapters.

Chapter 5: The Tactile Perception (TP) chapter addresses the method chosen in Chapter 3 to solve the first problem. This chapter expands on the method chosen and its functionality, followed by an evaluation of said method through experiments and their results. Finally, the functionality of the method is concluded.

Chapter 6: The PE chapter addresses the method chosen in Chapter 3 to solve the second problem. This chapter follows the same structure as Chapter 5.

Chapter 7: The In-Hand Manipulation (IHM) chapter addresses the method chosen in Chapter 3 to solve the third problem. This chapter follows the same structure as Chapter 5 and Chapter 6.

Chapter 8: The discussion chapter goes over the combined system and addresses problems, improvements, shortcomings and potential for future development.

Chapter 9: The conclusion chapter addresses the successfulness of the project.

Chapter 2

System Setup

The Shadow Dexterous Hand [25] is a sophisticated robotic hand with 24 DOF and a wide range of sensory feedback capabilities. To develop and test control algorithms for this complex system, simulation is an invaluable tool. In this chapter, the practical setup of our project is presented, which involves simulating the Shadow Dexterous Hand in Gazebo [26] within a Docker container [27].

This simulation is based on the Shadow Robot Company's [28] Robot Operating System (ROS) [8] packages [29], which provide a flexible and customizable framework for controlling the hand. This project uses Gazebo, a popular robot simulation environment, to simulate the physics of the hand and its interaction with the environment. To ensure reproducibility and portability, the development framework which encapsulates the simulation environment is built within a Docker container, which allows us to easily distribute our code and dependencies to other researchers.

In this chapter, a detailed description of the hardware and software used in this project's simulation setup is provided, which includes the software architecture of the simulation, including the ROS packages and Gazebo plugins used to simulate the hand and its environment.

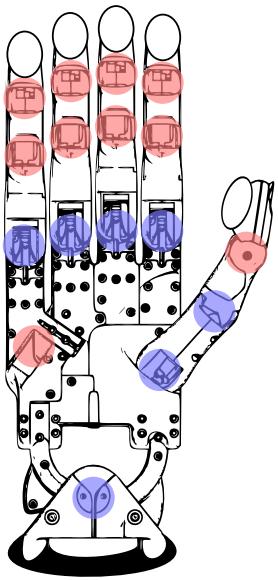
2.1 Simulation Setup

The Computer Aided Design (CAD) model of the Shadow Dexterous Hand is a highly detailed and accurate representation of the physical hand. The model is based on the original design of the real-world hand, which was developed by the Shadow Robot Company. The CAD model includes precise geometry for all of the hand's components, including the finger joints, tendons, and tactile sensors. The model also includes detailed material properties for each component, which are used to simulate the hand's physical behavior in the simulation.

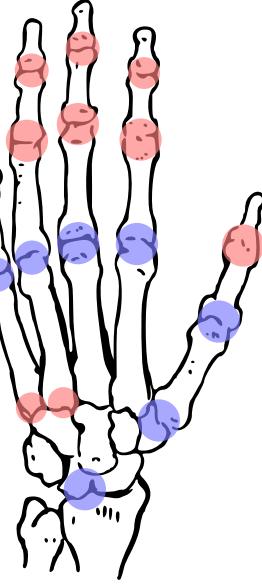
The structure of the hand can be seen in Fig. 2.1(a) compared to a human hand in Fig. 2.1(b). Here each joint is marked with a color, labeling the DOF for that particular joint. The red labels refer to joints with 1 DOF, while blue refers to joints with a DOF equal to 2. As seen here the Shadow Dexterous Hand provides human-like dexterity due to its similar kinematics, see Fig. A.1, and the comparable Range of Motion (ROM) of each joint, see Table A.1 and Table A.2.

The hand's geometry is modeled using a combination of standard shapes and custom-designed components. For example, the finger joints are modeled using a series of cylinders and spheres, which are connected by virtual tendons to simulate the motion of the real-world hand. The tactile sensors on the simulated hand, at the writing of this thesis, are purely aesthetical as representative simulated tactile sensors are yet to be supported as a standard component. To generate representative tactile data additional software is therefore required. The tactile sensors can be seen in Fig. 2.1(a) as the ellipsoids mounted at each fingertip.

Multiple hand configurations are available including a left hand, right hand, and both configurations mounted on Universal Robots (UR) manipulators [33]. The configuration chosen for this project is a Shadow Dexterous right hand without being mounted on a manipulator. Fig. 2.2 shows the CAD model of the Shadow Dexterous Hand in simulation.



(a) Shadow Dexterous hand with joints color coded depending on the degrees of freedom. The total number of degrees of freedom can here be seen as 24. This figure is based on [30].



(b) Shadow Dexterous hand with joints color coded depending on the degrees of freedom. The total number of degrees of freedom can here be seen as 25 [31]. This figure is based on [32].

Fig. 2.1: The Shadow Dexterous hand and a human hand red here marks a joint with 1 degree of freedom, while blue marks a joint with 2.

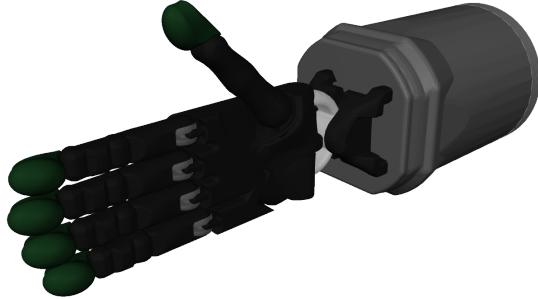


Fig. 2.2: A cutout of the simulated Shadow Dexterous Hand.

2.2 Software Setup

The software setup is shipped in a custom Ubuntu-based docker container [27, 34] with the necessary libraries to communicate and develop applications on the simulated as well as the physical hand, wrapped within a catkin workspace [35]. Additionally, the container comes with common-use libraries for Python and C++ development in ROS including `numpy`[36], `OpenCV` [37] and `dynamixel_reconfiguration` [38]. The development environment can be seen illustrated in Fig. 2.3. The communication between the provided ROS packages and the Gazebo simulator is achieved through the ROS-Gazebo framework.

When communicating with the Shadow Dexterous hand, the primary interface provided is the hand commander i.e. `SrHandCommander` which provides functionalities such as retrieving the current state of the hand, executing given path plans etc. To plan and execute a path \mathbf{Q} , which is computed as a list joint states $\mathbf{q}_i = [q_0, q_1, \dots, q_n]$ where $i \in \{0, 1, \dots, m\}$ with m being the number of states in the plan \mathbf{Q} , `SrHandCommander` parses the path to the `move_group` handled by MoveIt [39]. After receiving the plan, MoveIt goes through several evaluations to ensure

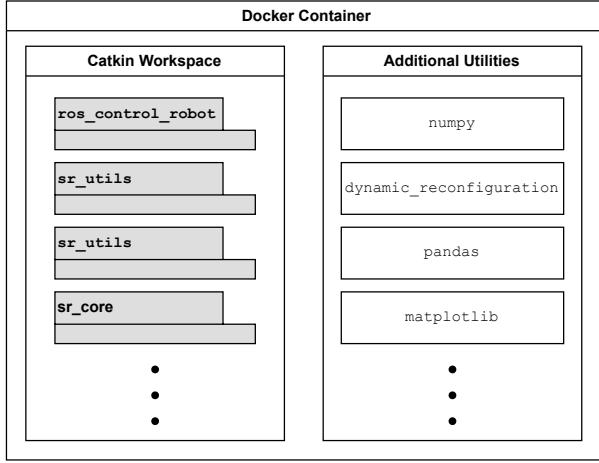


Fig. 2.3: The boxes marked with grey are ROS packages, while the white are python modules.

its validity, including state validation, collision checking, and motion planning. Before sampling a path, MoveIt first checks the validity of the start and goal states concerning the robot's joint limits, collision constraints, and other constraints like self-collision avoidance. This helps to ensure that the planned path is feasible and safe for the robot to execute. Once validated, the validated path $\mathbf{Q}_{\text{valid}}$ is parsed to The Open Motion Planning Library (OMPL) [40] where a safe path is built using sample-based path planning methods. Some examples of these methods include Probabilistic Roadmap Method (PRM), Fast Marching Tree (FMT) and Rapidly-exploring Random Tree (RRT)-Connect, the last of which is the default used in the Shadow Robotics configuration space position controller. Using these a full and valid path $\mathbf{Q}_{\text{full, valid}}$ is planned and parsed to the hand controller, which is build using the `ros_control` [41] framework. The `ros_control` package provides a hardware-agnostic interface, to enable the same controllers on multiple different robotics platforms, along with the same controller being applicable on real hardware as well as simulated hardware. Finally, the hardware interface communicates with a 24 Proportional Integral Derivative (PID) controllers, one for each joint to ensure control of each joint. The software architecture for communicating with the robotic hand can be seen in Fig. 2.4, which was inspired by figures from [42, 43, 44].

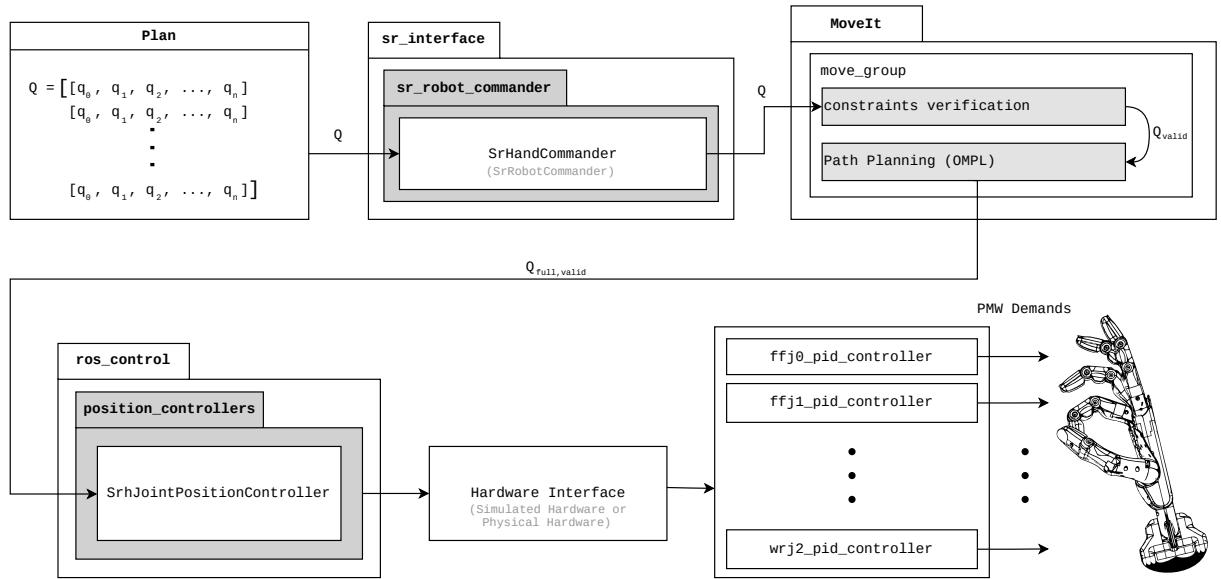


Fig. 2.4: Diagram showing the communication and control flow of interacting with the Shadow Dexterous hand.

Chapter 3

State of the Art

3.1 Problem 1 - Tactile Perception

Based on the contact model categories described in Chapter 4, the most representative was chosen to be soft finger (SF) models. Within the category of SF models, a method fit for this project's use case is to be chosen to solve problem 1. SF models can furthermore be divided up into three different categories: analytical elasticity-based models (AEBM), elastic foundation models (EFM), finite element models (FEM) [45] and machine learning (ML) models. The different categories can be seen organized in Fig. 3.1

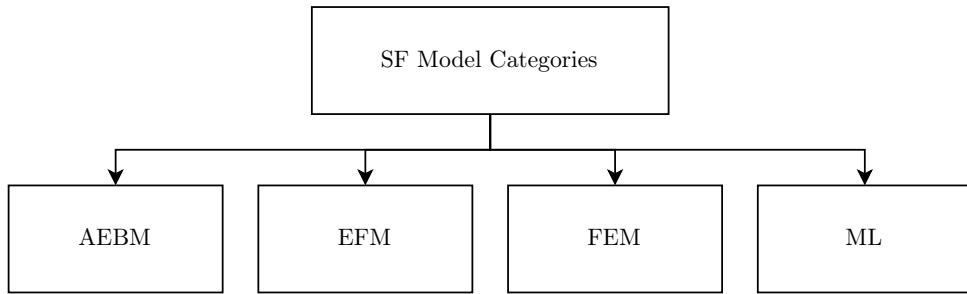


Fig. 3.1: Tree of methods for tactile perception.

AEBM are theoretical formulations of elastic contact areas and the stresses on both the surfaces and the sub-surfaces of the contacting bodies. The first of such models was introduced by Heinrich Rudolf Hertz in 1882 [46] and is still used for simple contact cases. In the formulation of the Hertzian contact model, two assumptions are made: Objects in contact are made of linear elastic materials and only small contact deformations occur compared to the dimension of an object. However, robotic EE fingertips are often made of non-linear elastic materials and for that reason, the Hertzian contact model does not represent the type of contact in this project [47, Chapter 37]. To improve on the Hertzian model, a more general formulation can be made which extends the model from linear to nonlinear elastic contacts [48, 49]. This power-law formulation subsumes the Hertzian contact theory while assuming a circular contact area. Other models have been proposed that combine the descriptions of both friction-contact and the shear-torsion as experienced by the bodies [50].

However, to more accurately describe the contacts involving robot fingers, viscoelastic soft contact model appear more relevant due to such fingers often being made of materials that show viscoelastic properties e.g., rubber, silicone and polymers. Simple models such as Kelvin-Voigt's [51] and Maxwell's [52] models describe the interaction between strain and stress as a spring-damper system in a serial or in a parallel configuration respectively. Models which expand on this idea describe the reacting force as the product of the temporal and the elastic response while incorporating previous stress responses [53]. To simplify this formulation alternatives have been developed to assume no past stress [54, 55, 56]. Upon these, more modern techniques have been developed which have seen use in similar use cases as the ones of interest in this project. One method attempts to expand the description of contacts between rigid indentors and elastic half-spaces, using the Matrix Inversion Method (MIM) as introduced by Kalker [57], to viscoelastic half-spaces as well. Assuming the surfaces are frictionless, the relationship is described in terms of the pressure distribution, the resultant force on the indenter and the penetration [58]. Attempts involving solutions to Boussinesq's problem for polynomial pressures acting over polygonal domains [59]

have also been developed and modernized by combining it with Cerruti's solution [60]. However due to numerical singularities being present, modifications are made to threshold the model. For a more complete description without singularities, Love's formulation has been added leading to a more accurate analytical representation but with the cost of an increased computational complexity [61]. For these Boussinesq-based approaches to be representative two assumptions are made 1) There exists a linear relationship between stress and strain, referred to as deformation, and 2) strains are infinitesimal [62, Chapter 6].

EFM are methods developed to build upon AEBM by allowing a simple discrete contact calculation in more general surface geometries. Here the deformable part of the contact is modeled as a layer over a rigid base with a series of discrete and independent springs in the contact normal. A widely used example of this method is Winkler's elastic foundation model [63], which has been used in structural engineering for modeling different properties of beams such as stability [64], vibrations and buckling [65]. Other EFM methods have shown accurate modeling performance when applied within the field of medical engineering. Here a comparative study between AEBM, EFM and FEM demonstrate the suggested modified EFM performs better than the alternatives in 3D knee models when predicting prosthetic knee performance [45]. A different method attempts to attain vivo contact pressure predictions for improved knee replacement designs [66] Within the field of robotics EFM have provided solutions to problems such as slip [50], compliance, sliding [67, 68], stiffness and contact mechanics [69] of anthropomorphic grippers. One such method derives friction constraints based on general expressions for non-planar contacts of elastic bodies, where the local geometry and structure of the objects in contact are taken into account. Using these, a linear complementary problem is formulated and solved, resulting in the normal and frictional forces applied at each contact, as well as the relative velocity of the bodies involved [70].

FEM are popular general tools for solving PDE [71] and have seen contact applications in a wide range of engineering disciplines due to the assumptions made in AEBM and EFM not being applicable in these cases. A great number of these cases exist within the manufacturing industry [72] whereas one example is the metal forming processes. Specifically, the estimation of wheel-rail profiles [73] has been addressed using FEM due to the estimation of contacts over a greater surface is needed than what is assumed in AEBM and EFM. Other applications such as quality control through sliding wear estimation [74], analysis of the responses of fully coupled thermo-elasto-plastic solids in contact [75] and performing diagnostics of failures in induction motors [76]. Due to the complexity of modeling the contacts within robotics, FEM have become a popular choice and enabled tactile applications such as cobots tactile skin for ensuring collaborative behavior when in contact [77], performance estimation of new tactile sensor technologies [78] and evaluating complex contact types by extending simulations and analysis systems [79]. The modeling complexity has furthermore inspired using FEM as ground truth results when synthesizing ML data in simulations for deep learning models, which has enabled execution speeds 75 times greater than simply evaluating FEM [80, 81, 82].

The use of these ML models has enabled realistic simulations of tactile sensor data. Current literature applies DL-based approaches to simulate tactile sensor data for various tasks [83, 84]. For instance, simulating realistic tactile images from simulated contact depth to bridging the reality gap for vision-based tactile sensing using a diffusion model [85]. Similarly, a conditional Generative Adversarial Network (cGAN) has been used to simulate realistic tactile sensory data for use in tactile tasks [85]. Solutions using DL models purely based on Multi Layered Perceptron (MLP) have been applied to enable real-time simulated realistic tactile data [86].

Given the methods presented above, the AEBM Boussinesq-Cerruti approach is considered along with the MLP based DL model.

Although the Boussinesq-Cerruti approach can produce precise tactile data and can be tailored to suit a particular case, it faces certain challenges. The model relies on certain assumptions regarding the materials in contact, including linear deformation and infinitesimal strains. Furthermore, evaluating the model requires complex calculations, such as multidimensional integrals, which significantly increase computation time and hinder real-time

performance. In contrast to the transparency offered by the Boussinesq-Cerruti approach, the MLP based approach is limited by the black-box nature of DL models. Despite this drawback, MLP based DL models offer several benefits, such as low execution time and high adaptability to complex systems. Due to the high adaptability and option for real-time performance, the DL model presented in [86] is chosen to solve the tactile perception problem i.e. problem 1.

3.2 Problem 2 - Pose Estimation

PE, which involves determining the position and orientation of an object in 3D space, has been the subject of many research studies. The literature has identified two main categories of methods for solving this problem: those based on DL, and those based on point cloud registration.

These can along with their subcategories be seen in Fig. 3.2 as inspired by [87].

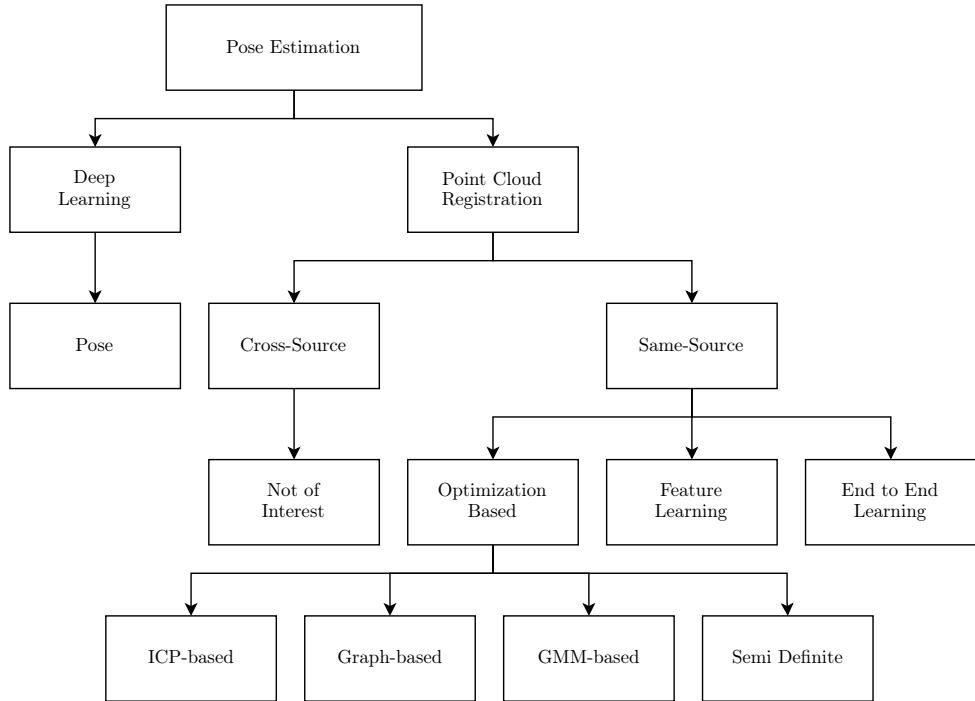


Fig. 3.2: Tree of methods for solving the point cloud registration problem. The categorization is inspired by [87].

Purely DL based methods learn feature representations of input data, often in the form of images, and use them to estimate the subject's pose. This is commonly done in the context of human pose estimation [88, 89, 90]. While these methods have shown extensive use in these cases, their applicability in this project is limited and thus excluded from consideration.

The other method group i.e. point cloud registration (PCR) methods are separated into two subgroups: cross-source and same-source point cloud (PC)s. Here cross-source refers to a PC produced by combining information from sensors of different kinds e.g. visual- and tactile sensors, while same-source methods only produce PCs based on information from the same kind of sensors e.g. only tactile sensors. While cross-source approaches have shown utility in an extensive range of applications [91, 92, 87] their applicability in this project is minimal, as purely tactile pose estimation is the problem of interest as presented in Chapter 1.

PCR methods from the same source data can be categorized into three sub-categories: end-to-end learning, feature-learning, and optimization-based methods. End-to-end learning-based methods use a neural network to estimate

the transformation matrix that aligns two point clouds. Proposed solutions include using neural networks for scene completion to estimate the relative pose between RGB-D scans [93], learning registration patterns as parametric functions through a scan completion module and pairwise matching module [94], and a fast feature-metric point cloud registration framework to minimize the feature-metric projection error without correspondences [95].

In contrast, feature-learning methods use deep neural networks to learn robust feature correspondence searches, which are then used in estimation algorithms such as Random Sample Consensus (RANSAC). In the literature, models have been developed to extract local geometric descriptors from RGB-D reconstructions [96], to learn globally informed 3D local feature descriptors [97], and to use siamese deep learning architectures with convolutional layers through a voxelized smoothed density value (SDV) representation [98].

Lastly, registration methods based on optimization are employed to estimate the transformation matrix through two stages: correspondence searching and transformation estimation. Their goal is to minimize a cost function that gauges the dissimilarity between two point clouds. Within this category, there are four sub-categories identified: Iterative Closest Point (ICP)-based, graph-based, Gaussian Mixture Model (GMM)-based, and semi-definite programming-based methods.

Since the original proposal in 1992 [99] using point-to-point correspondences, ICP-based methods have evolved and incorporated different types of correspondences to improve performance. Examples include point-to-plane [100] and plane-to-plane [101]. Modern approaches also employ complementary methods such as point cloud filtering, adaptive fireworks algorithms, and KD-Trees [102].

The main idea of graph-based registration methods is to use a non-parametric model [103]. In this method, correspondences between two graphs are found by considering both the vertices and edges, making it an optimization problem [103]. To solve this optimization problem, there are two categories of graph-matching methods based on the objective functions' constraints: second-order and high-order methods [104]. Second-order methods include Cross-Source Graph Matching (CSGM) [91], which uses a linear program to solve the graph-matching problem and apply it to solve the cross-source point cloud registration task, Factorized Graph Matching (FGM) [105] factorizes the large pairwise affinity matrix into smaller matrices and solves the graph-matching problem with a simple path-following optimization algorithm. Spectral graph [106] uses a spectral relaxation method to approximate the Quadratic Assignment Problem (QAP), and Semi-Definite Programming (SDP) relaxation is used to relax the non-convex constraint using a convex semi-definite. While higher-order graph matching provide methods for [107] design a probabilistic approach to solve the high-order graph-matching problem, while [108] design a triangle similarity and convert the graph-matching problem into a tensor optimization problem. More recent work, such as [109] suggests an elastic net to control the trade-off between the sparsity and accuracy of the matching results by incorporating the Elastic-Net constraint into the tensor-based graph matching mode.

GMM-based methods commonly tackle the point cloud registration problem by transforming it into a likelihood maximization problem for the input data. This has resulted in the development of several optimization strategies aimed at maximizing the likelihood and optimizing the transformation matrix. For instance, a motion drift idea was introduced into the GMM framework by [110] in the form of Coherent Point Drift (CPD) which imposes constraints on transformation estimation. In another approach,[111] combines GMM with the convex hull to reduce computation complexity. Furthermore, Joint Registration of Multiple Point Sets (JRMPC) [112] cast registration as a clustering problem where the transformation is optimized by solving the GMM. Recently, Deep Gaussian Mixture Registration (DeepGMR) [113] employed DL to learn the correspondences between GMM components and points, enabling the estimation of both the transformation and GMM parameters in a single forward step.

Lastly, within semidefinite programming different optimization groups exist, such as Second Order Cone Programming (SOCP), Quadratic Programming (QP) and Quadratically Constrained Quadratic Programming (QCQP). Due to the subject of interest being a rotation in $\text{SO}(3)$, the constraints of the rotation matrix, i.e. quadratic constraints,

must be respected. Because of this, the methods of QCQP are of interest. One such example provides estimates which are insensitive to a large fraction of spurious correspondences through decoupling the scale, rotation, and translation estimation. This decoupling enables the solving of these in cascade for the three transformations. The method is referred to as TEASER (Truncated least squares Estimation And SEmidefinite Relaxation), which solve large SDP relaxations, and additionally comes with a second fast and certifiable algorithm, named TEASER++. To decrease execution time this method uses Graduated Non-Convexity (GNC) to solve the rotation subproblem and applies Douglas-Rachford Splitting to enable efficiently certify global optimality [114]. Secondly, Invariant-based Highly Robust Point Cloud Registration (IRON) applies a similar methodology as to TEASER, but instead applies RANSIC (RANdom Samples with Invariant Compatibility) to robustly estimates the scale between two sets of point clouds [115]. Finally, Relaxed Convex Quadratic Programming (RCQP) formulates a QCQP problem with a full set of quadratic rotational constraints and obtains a Lagrangian dual relaxation, which empirically recovered a globally optimal solution in 100 % of the tested cases, although why strong relaxation seems to hold has yet to be shown [116].

Among the categories presented above, the ones of particular interest are optimization-based techniques due to their mature mathematical foundation and possible certifiable optimality and outlier rejection capabilities, and DL-based techniques due to their adaptability and low execution time. While DL-based methods can learn feature representations of point clouds and estimate the transformation between two point clouds, optimization-based approaches with outlier rejection can effectively handle noise and outliers in the data, which is a common problem within the PCR problem. Due to this, the chosen method is the optimization based RCQP method with GNC outlier rejection [117].

3.3 Problem 3 - In-Hand Manipulation

References:

resources:

Chapter 4

Modeling

To model the contact between the EE's tactile sensors, eight different categories exist as identified in [118]. The three most common ones within the field of robotics [47, Chapter 37] are point-contact-without-friction (PwoF), hard finger (HF) and the SF model as shown in Fig. 4.1.

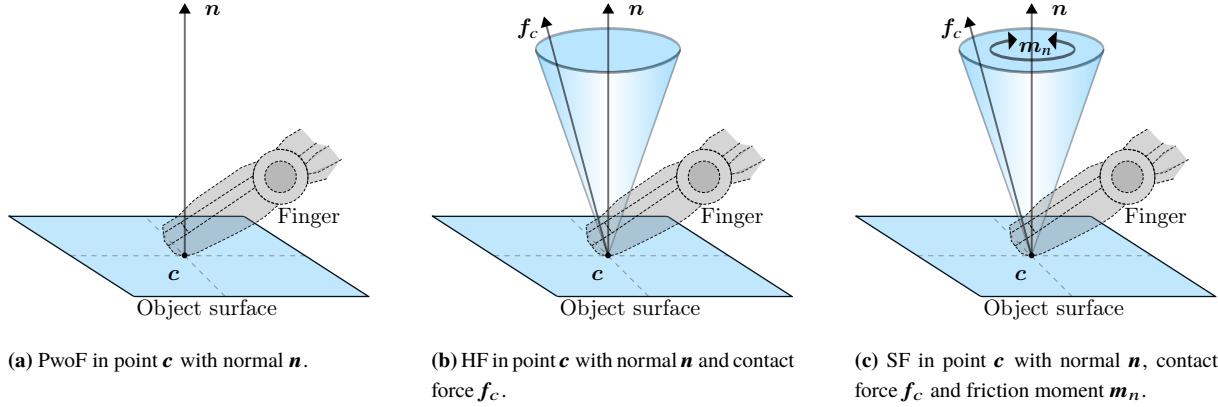


Fig. 4.1: The three most commonly used contact models.

The PwoF model, as shown in Fig. 4.1(a), can only represent forces along the surface normal $\mathbf{n} \in \mathbb{R}^3$ at the point of contact $\mathbf{c} \in \mathbb{R}^3$ and thus the model does not support surface deformations between the two contacting objects. This model is applied in cases where very little deformation is present, along with the contact having a friction coefficient approximately equal to zero [47, Chapter 38].

The HF model, as shown in Fig. 4.1(b), is representative when the friction between objects is significant, while the contact deformation is small enough to ignore friction moments and deformations [47, Chapter 38]. To model the friction acting on the contact point a great number of methods exist, a common one being the Coulomb friction with different modifications depending on the use case [119]. This model states that the frictional force acting on an object can be formulated as

$$f_f = f_N \mu, \quad (4.1)$$

with f_f being the magnitude of the Coulomb friction, f_N being the magnitude of normal force in the point of contact and $\mu \in [0, 1]$ being the friction coefficient. One visualization of this linear relationship can be seen in the cones illustrated in Fig. 4.1(b) and Fig. 4.1(c). These cones are referred to as friction cones, which for a hard finger model can be formulated as

$$C_{f,\text{HF}} = \{ f_c \mid f_t \leq \mu f_z, \mu f_z \geq 0 \} \quad , \quad f_t = \sqrt{f_x^2 + f_y^2}. \quad (4.2)$$

Here f_c is the magnitude of the contact force, f_t is the magnitude of the tangential force. f_x , f_y and f_z are the magnitudes of the x , y and z components of the contact force ($f_c \in \mathbb{R}^3$) and μ is the friction coefficient [47, Chapter 37]. By applying a contact force that ensures the friction stays greater than the magnitude of the tangential force, neither object slips i.e. f_z must ensure that $f_z \mu$ stays greater than f_t for the objects not to slip. Visually this is the case when the contact force f_c stays within the friction cone, which enables a friction-based grasp type referred to as force closure. Specifically, force closure refers to when the composite wrench cone contains the entire wrench space so that any external wrench w_{ext} on the body can be balanced by contact forces [120]. A force that commonly contributes significantly to the external wrench, and thus to the tangential force, is gravity.

The SF model, as shown in Fig. 4.1(c), is used to represent scenarios where both friction and surface deformations are significant. Due to deformations of the finger, an additional torsional moment about the contact normal will be present [47, Chapter 38]. While an analytical formulation of the SF relation depends on the pressure distribution inside the contact, and can only be derived for a limited number of special cases, the general case can be approximated using

$$\mathcal{E}_{f,SF} = \left\{ f_c \mid f_t^2 + \frac{m_n^2}{e_n^2} \leq \mu^2 P^2 \right\} , \quad f_t = \sqrt{f_x^2 + f_y^2}. \quad (4.3)$$

This formulation forms a contact ellipsoid $\mathcal{E}_{f,SF}$ which describes the relationship between the tangential force $f_t \in \mathbb{R}^3$ and friction moment $m_n \in \mathbb{R}^3$. The friction parameters in this expression remain the same as for the friction cone, with the additional m_n being the magnitude of the frictional moment, e_n being the eccentricity parameter i.e. the height of the aforementioned ellipsoid and P being the magnitude of the pressure applied from the contact point along the contact normal \mathbf{n} [68, 70].

Based on the model categories described above, the most representative for this project's case, are the SF models since these can provide information about the contact surface's shape, thus enabling the reconstruction of the contact shape from the application of a force distribution [121] i.e. the Inverse Elasticity Problem (IEP). Additionally, these models support descriptions of friction which is crucial to manipulate objects in hand. Illustrations of the system as a SF with friction cone, pressure distribution and the enabling of force closure can be seen in Fig. 4.2 and Fig. 4.3 respectively.

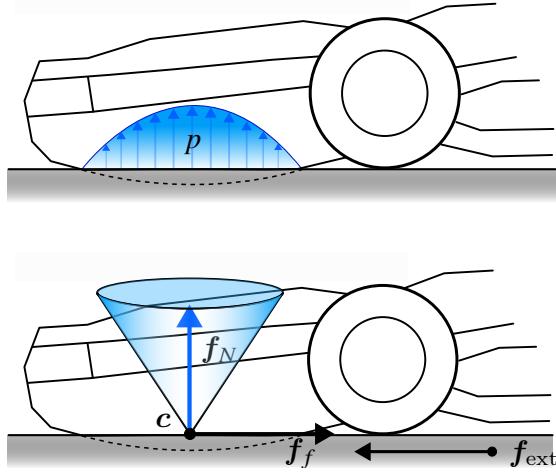


Fig. 4.2: The pressure distribution p and friction cone of a SF model experiencing an external force f_{ext} .

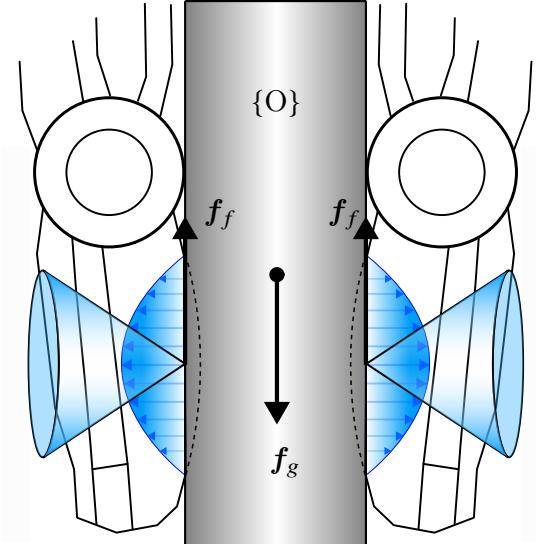


Fig. 4.3: the pressure distribution and friction cone causing force closure to prevent the object $\{O\}$ from falling due to the gravitational force f_g .

When modeling the kinematics of an anthropomorphic gripper with frame $\{H\} \in \mathbb{R}^{4 \times 4}$ in world frame $\{W\} \in \mathbb{R}^{4 \times 4}$ interacting with an object with frame $\{O\} \in \mathbb{R}^{4 \times 4}$ the relevant parameters must be addressed. In this system the object with position $\mathbf{p} \in \mathbb{R}^3$ and pose $\mathbf{u} \in \mathbb{R}^6$, with the orientation either being represented as a four-dimensional quaternion or a three-dimensional Euler angle, makes contact with the gripper in points $\mathbf{c}_i \in \mathbb{R}^3$. These contact points have frames $\{C_i\} \in \mathbb{R}^{4 \times 4}$ with axes $\{\mathbf{n}_i, \mathbf{t}_i, \mathbf{o}_i\} \subset \mathbb{R}^3$, where $\mathbf{n}_i \in \mathbb{R}^3$ points perpendicular to the contact plain towards the object, while the remaining are contained within the contact plane. For each of these parameters $i = 1, 2, \dots, n_c$, where n_c is the number of contact points. The twist of $\{O\}$ described in $\{W\}$ is denoted $\mathbf{v} = [\mathbf{v}^\top \ \mathbf{w}^\top]^\top \in \mathbb{R}^6$ while the non-contact wrench i.e. the wrench caused by external forces such as collisions with the environment and gravity, is $\mathbf{w} = [\mathbf{f}^\top \ \mathbf{m}^\top]^\top \in \mathbb{R}^6$. The gripper's state is described in terms of its

joints, of which it has n_q , named $\mathbf{q} = [q_1 \ q_2 \ \dots \ q_{n_q}]^\top \in \mathbb{R}^{n_q}$ each of which is revolute and can exert a torque $\boldsymbol{\tau} = [\tau_1 \ \tau_2 \ \dots \ \tau_{n_q}]^\top \in \mathbb{R}^{n_q}$. These parameters can be seen illustrated in Fig. 4.4 showing the system model. While only a single finger here is illustrated the naming conventions and representations simply scale to all the

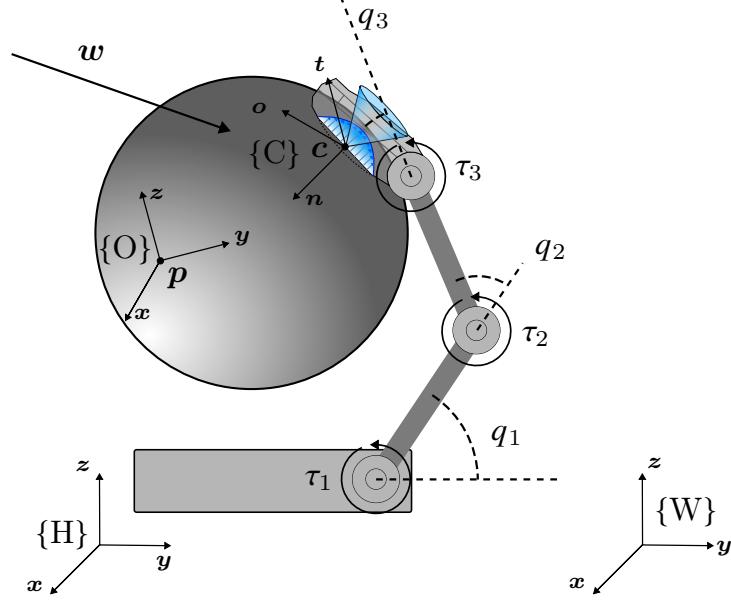


Fig. 4.4: The model of the world representation for this project.

EE's DOFs.

In this system, the twists and wrenches of a contact point c_i on the object and hand, given in contact frame $\{C\}_i$ is referred to as $\mathbf{v}_{i,\xi} \in \mathbb{R}^6$ and $\mathbf{w}_{i,\xi} \in \mathbb{R}^6$, with $\xi = \{\text{obj}, \text{hnd}\}$. Given multiple contact points, complete vectors of twist and wrench can be expressed by appending each contact point's twist and wrench vector. These contain all twists and wrenches of the grasp, one for the object and one for the hand. These vectors are referred to as

$$\mathbf{v}_{c,\xi} = [\mathbf{v}_{1,\xi}^\top \ \mathbf{v}_{2,\xi}^\top \ \dots \ \mathbf{v}_{n_c,\xi}^\top]^\top \in \mathbb{R}^{6 \cdot n_c} \quad (4.4)$$

and

$$\mathbf{w}_{c,\xi} = [\mathbf{w}_{1,\xi}^\top \ \mathbf{w}_{2,\xi}^\top \ \dots \ \mathbf{w}_{n_c,\xi}^\top]^\top \in \mathbb{R}^{6 \cdot n_c} \quad (4.5)$$

respectively.

These definitions are used to describe and analyze the kinematics of grasping and the parameters involved in holding and manipulating objects in hand, also referred to as grasp kinematics (GK). Within GK two matrices are of special interest: the grasping matrix \mathbf{G} and the hand Jacobian \mathbf{J} . The grasping matrix describes the transformation between the twist or wrench of the object in world frame $\{W\}$ to the twists or wrenches of the object in contact frames $\{C\}_i$. The grasp matrix thus can be expressed as

$$\mathbf{G} = [\mathbf{G}_1 \ \mathbf{G}_2 \ \mathbf{G}_3 \ \dots \ \mathbf{G}_{n_c}], \quad (4.6)$$

where $\mathbf{G}_i \in \mathbb{R}^{6 \times 6}$ describes the transformation from $\{W\}$ to the individual $\{C\}_i$, and thus $\mathbf{G} \in \mathbb{R}^{6 \times 6 \cdot n_c}$ describes the transformations for all contact points. Using this grasp matrix, the object wrench and twist can be computed in all contact frames as

$$\mathbf{v}_{c,\text{obj}} = \mathbf{G}^\top \mathbf{v} \quad \text{and} \quad \mathbf{w}_{c,\text{obj}} = \mathbf{G}^\top \mathbf{w}. \quad (4.7)$$

While the grasp matrix describes the transformation from $\{W\}$ to object contact frames, the hand Jacobian relates the joint velocities and torques to the contact twists and wrenches on the hand. The hand Jacobian can thus be

expressed as

$$\mathbf{J} = [\mathbf{J}_1^\top \mathbf{J}_2^\top \mathbf{J}_3^\top \cdots \mathbf{J}_{n_c}^\top]^\top, \quad (4.8)$$

for all contact points. Here $\mathbf{J}_i \in \mathbb{R}^{6 \times n_q}$ for $i = 1, 2, \dots, n_c$ are the individual contact point's hand Jacobians and thus $\mathbf{J} \in \mathbb{R}^{6 \cdot n_c \times n_q}$ is the complete. Using the complete hand Jacobian, the contact twists and wrenches on the hand can be related to the joint velocities and torques as

$$\boldsymbol{\nu}_{c,\text{hnd}} = \mathbf{J}\dot{\boldsymbol{q}} \quad \text{and} \quad \boldsymbol{\tau} = \mathbf{J}^\top \boldsymbol{w}_{c,\text{hnd}}. \quad (4.9)$$

The modeling described above will enable the use of methods for solving the presented problems. These methods will be determined in Chapter 3.

Chapter 5

Tactile Perception

To solve problem 2 and 3, the tactile perception solution must provide estimates of contact positions, contact normals and skew forces.

This chapter presents an analysis of the performance of different techniques to solve these problems, including a DL model for simulating realistic tactile skew forces, and Recursive Least Squares (RLS) for estimating contact normals. Contact normals are essential for accurately estimating the pose of an object in contact, while skew forces are critical for predicting the behavior of an object when it is grasped and manipulated by the Shadow Dexterous hand.

Firstly, the technique behind estimating the skew forces is a DL model, which includes its architecture, is described, and the methodology used to test the network is presented. The testing methodology involves the use of various input data, and the output is analyzed for accuracy and realism. The findings are presented and discussed, including the strengths and weaknesses of the network in simulating tactile data. Finally, an assessment is made of the network's ability to produce tactile data that is realistic.

The same procedure is applied for the RLS normal estimation.

The force and normal estimates are compared to the ones provided by Gazebo's physics engine, and the known Ground Truth (GT) from where conclusions are drawn.

The software used in this chapter is a regression neural network implemented as a Gazebo ModelPlugin [122] in C++. However, the DL model plugin used in the original publication [86] has not been updated since 2018, making the code incompatible with the current version of Gazebo API. Moreover, the licensing issues with the files in the `xmlrpc++` library, which were used for base64 encoding and decoding, necessitated their removal [123]. To address these issues, each has been resolved and the plugin has been reorganized and repackaged for compatibility with the current version of Gazebo. The original version of the plugin can be found in [124], while the fixed and updated version is available at [125].

The availability of the updated plugin ensures that the project can continue to benefit from the capabilities of the MLP based DL model for simulating realistic tactile data in the current version of Gazebo.

5.1 Methods

5.1.1 Recursive Least Squares

To use RLS to estimate the contact normal, we require the estimated linear velocity $\hat{v} \in \mathbb{R}^3$. However, since the contact points may not be consistent across the surface during motion, some points may be missing at certain time steps. To address this issue, the centroid of the contact points $\bar{c}_t \in \mathbb{R}^3$ is used as a representative value for each time step. Therefore, we can compute the estimated linear velocity as

$$\hat{v} = \frac{\bar{c}_t - \bar{c}_{t-1}}{\Delta t}, \quad (5.1)$$

where $\bar{c}_t \in \mathbb{R}^3$ is the centroid of contact points in timestep t , \bar{c}_t is the centroid for the previous timestep $t - 1$ and Δt is the time between samples, which in this case is 0.01 s as the sampling frequency is 100 Hz. The velocity at

the fingertip in contact will then be computed as

$$\mathbf{v}_{tip} = \hat{\mathbf{v}} + [\boldsymbol{\omega}]_{\times} \mathbf{R}_{tcp}^{base} \mathbf{r}, \quad (5.2)$$

where $\hat{\mathbf{v}}$ is the estimated linear velocity computed above, $[\boldsymbol{\omega}]_{\times}$ is the skew-symmetric matrix of the angular velocity $\boldsymbol{\omega}$ i.e.

$$[\boldsymbol{\omega}]_{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \quad \boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z], \quad (5.3)$$

$\mathbf{R}_{tcp}^{base} \in \mathbb{R}^{3 \times 3}$ is the rotation matrix from the hand's base to its Tool Center Point (TCP), and $\mathbf{r} \in \mathbb{R}^3$ is the position of the contact point in the base frame.

To compute the normal estimate, additionally, an initial guess is needed which is computed by

$$\hat{\mathbf{n}}_0 = \frac{\hat{\mathbf{v}}_1 \times \hat{\mathbf{v}}_0}{\|\hat{\mathbf{v}}_1 \times \hat{\mathbf{v}}_0\|_2}, \quad (5.4)$$

where $\hat{\mathbf{n}}_0 \in \mathbb{R}^3$ is the initial estimate, $\hat{\mathbf{v}}_0$ and $\hat{\mathbf{v}}_1$ are the linear velocity estimates for time step 1 and 0, and $\|\cdot\|_2$ is the ℓ_2 -norm.

Using the quantities above, the two main components $\mathbf{L}_n^1 \in \mathbb{R}^{3 \times 3}$ and $\mathbf{L}_n^2 \in \mathbb{R}^{3 \times 3}$ of the desired estimate $\hat{\mathbf{n}}_{new} \in \mathbb{R}^3$ can be found. Firstly, \mathbf{L}_n^1 is computed as

$$\mathbf{L}_n^1 = \mathbf{L}_n^1 - \beta \Delta t \mathbf{L}_n^1 + \frac{\Delta t K_L}{1 + \|\mathbf{v}_{tip}\|_2^2} \mathbf{v}_{tip} \mathbf{v}_{tip}^T, \quad (5.5)$$

with $\beta \in \mathbb{R}$ being a decay factor, $K_L \in \mathbb{R}$ is a gain and $\|\cdot\|_2^2$ is the squared ℓ_2 -norm.

The second component \mathbf{L}_n^2 is computed by

$$\mathbf{L}_n^2 = \mathbf{L}_n^2 - \beta \Delta t \mathbf{L}_n^2 + \frac{\Delta t K_L}{1 + \|\boldsymbol{\nabla}\|_2^2} \boldsymbol{\nabla} \boldsymbol{\nabla}^T, \quad (5.6)$$

where $\boldsymbol{\nabla} \in \mathbb{R}^3$ is the cross product between the contact force \mathbf{f}_c and the velocity of the tip \mathbf{v}_{tip} i.e.

$$\boldsymbol{\nabla} = \mathbf{f}_c \times \mathbf{v}_{tip}. \quad (5.7)$$

Using these components a Proportional-Derivative (PD) controller is used to compute the change in normal $\dot{\mathbf{n}} \in \mathbb{R}^3$ as

$$\dot{\mathbf{n}} = -(\gamma_1 \mathbf{L}_n^1 + \gamma_2 \mathbf{L}_n^2) \mathbf{n}, \quad (5.8)$$

where γ_1 is the proportional gain, γ_2 is the derivative gain and \mathbf{n} is the current estimate. By computing the cross product between \mathbf{n} and $\dot{\mathbf{n}}$, the angular velocity $\boldsymbol{\omega}$ is computed

$$\boldsymbol{\omega} = \mathbf{n} \times \dot{\mathbf{n}}. \quad (5.9)$$

Finally, the new normal estimate $\mathbf{n}_{new} \in \mathbb{R}^3$ can be computed as

$$\mathbf{n}_{new} = e^{\left[\frac{\Delta t}{2} [\boldsymbol{\omega}]_{\times} \right]} \mathbf{n}, \quad (5.10)$$

where $e^{\left[\frac{\Delta t}{2} [\boldsymbol{\omega}]_{\times} \right]} \in \mathbb{R}^{3 \times 3}$ is the exponential map of the skew-symmetric matrix represents the rotation due to the angular velocity over the time step.

5.1.2 Network Architecture

In [86] two different architectures were built, where architecture B is chosen due to its better greater accuracy and lower execution time. The network architecture B can be seen in Fig. 5.1. As inputs, the network takes one

contact position $\mathbf{c} = [c_x, c_y, c_z]$ along with three force vector samples $\mathbf{f}_1 = [f_{1,x}, f_{1,y}, f_{1,z}]$, $\mathbf{f}_2 = [f_{2,x}, f_{2,y}, f_{2,z}]$ and $\mathbf{f}_3 = [f_{3,x}, f_{3,y}, f_{3,z}]$, and a temperature input $T \in \mathbb{R}$, which makes the input $[\mathbf{c}, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, T] \in \mathbb{R}^{13}$. The outputs of the network are the data format produced by the physical sensor i.e. an output vector of $[pdc, pac, tdc, tac, e_1, \dots, e_{19}] \in \mathbb{R}^{23}$, where pdc is the pressure DC signal, pac is the pressure AC signal, tdc is the temperature DC signal, tac is the temperature AC signal and e_1 to e_{19} are the electrode activations.

The network architecture consists of four MLPs, one for interpreting the position input, one for interpreting the force inputs, one for interpreting the temperature input, and one for combining the interpretations of force and position inputs. MLP 1, responsible for interpreting the position data, consists of four hidden layers, three of which contain 512 neurons and uses Rectified Linear Unit (ReLU) as the activation function, while the fourth and last uses a linear activation function with 64 neurons. The activation functions can be seen marked red if they use ReLU, green if they use a linear activation function and blue if they use a sigmoid activation function.

MLP 2 interprets the force inputs but rather than using 512, uses 256 for its hidden layers, while still having the linear activation function and the 64 neurons in its fourth layer. This MLP further differs as the 256 neuron layers apply ℓ_1 bias regularization. MLP 3 produces a temperature correction vector using two hidden layers, one with 256 neurons and a sigmoid activation function and one with 23 neurons and a linear activation function. The last MLP, MLP 4 takes in the element-wise product of MLP 1 and 2, parses the product through two 256 neuron layers with ReLU and one 23 neuron layer with a linear activation function.

The products of MLP 3 and 4 are summed and parsed as the model's output.

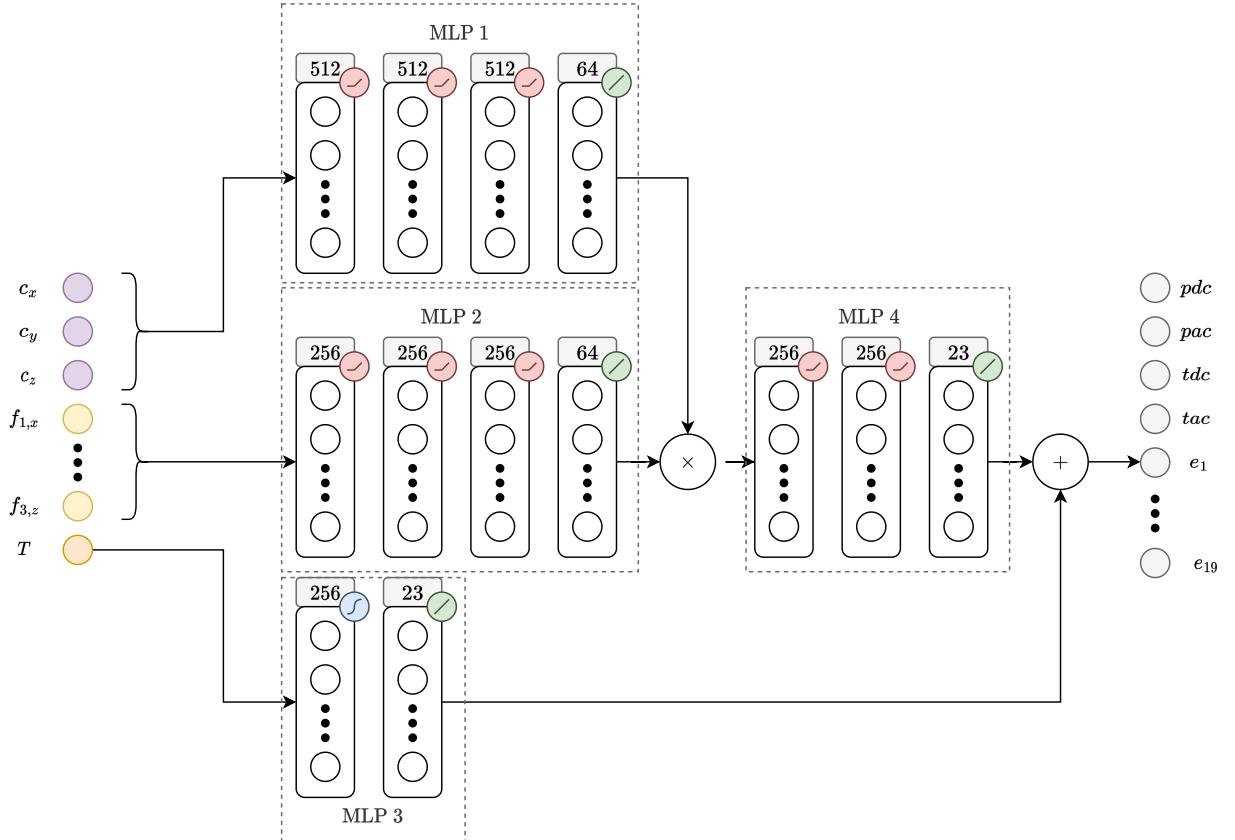


Fig. 5.1: DL model B from [86], which also has provided inspiration for this illustration.

5.1.3 Network Training Procedure

The model presented in 5.1.2 Network Architecture was trained using a custom dataset collected by the authors. The dataset D contains $N_{dp} = 300\,000$ tactile sensor readings, consisting of the complete BioTac sensor data as well as the applied reference forces. This can be seen below

$$D = \begin{bmatrix} 0 & pcd & pac & tdc & tac & e_1 & \cdots & e_{19} & f_{1,x} & \cdots & f_{3,z} & c_x & c_y & c_z \\ 1 & pcd & pac & tdc & tac & e_1 & \cdots & e_{19} & f_{1,x} & \cdots & f_{3,z} & c_x & c_y & c_z \\ \vdots & & & & & & & & \vdots & & & & & \\ N_{dp} & pcd & pac & tdc & tac & e_1 & \cdots & e_{19} & f_{1,x} & \cdots & f_{3,z} & c_x & c_y & c_z \end{bmatrix} \in \mathbb{R}^{N_{dp} \times 35} \quad (5.11)$$

All inputs are normalized to have zero mean and unit variance, based on the distribution of the captured data. The three force vectors are sampled at intervals of 100 ms to prevent overfitting and avoid unrealistic reactions to high-frequency inputs that the physics simulator cannot replicate. The BioTac sensor electrode values exhibit a non-linear dependence on device temperature and attempts to correct this before inputting the data resulted in poor performance. Instead, the network was trained to compensate for this dependency by itself. During simulation, a constant temperature was assumed, which is typically the average temperature of the training set. The network generates simulated electrode and pressure signals as outputs but does not currently simulate temperature outputs.

The forces were collected using a calibrated six-axis force-torque sensor [126] with a nominal force resolution better than 0.01 N.

The contact position is reconstructed optically using a calibrated HD webcam and two AprilTag markers [127], one mounted on the BioTac and one attached to the probe object. The setup for this can be seen in Fig. 5.2. Once the contact positions were collected, optimization-based calibrations were made to gain more accurate position estimates.

The data set has been made publicly available and can be found in [128].

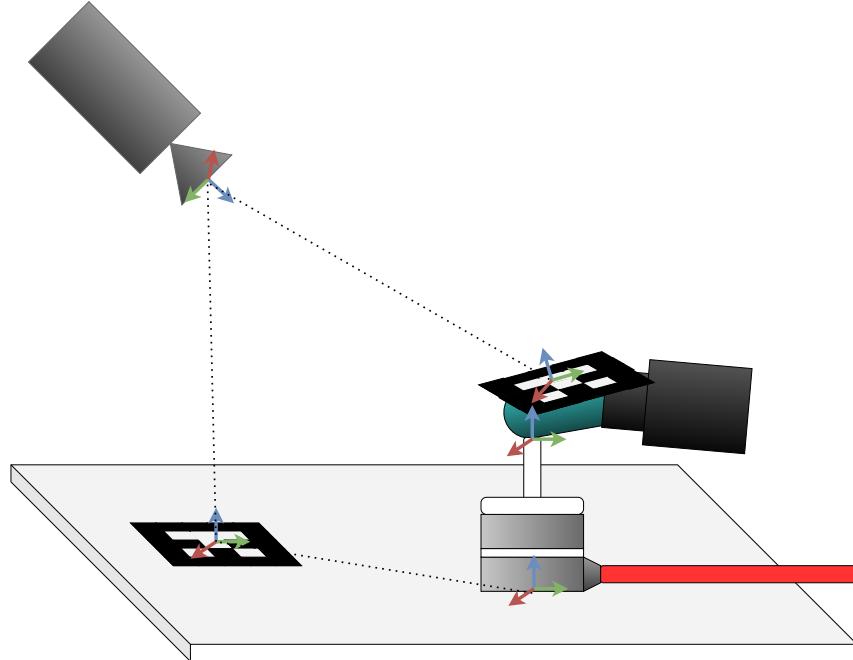


Fig. 5.2: Experimental setup for gathering data to train DL model B, as inspired by [86].

5.2 Experimental Setup

5.2.1 Contact Normal Estimation

To test the RLS method's ability to estimate contact normals the index finger makes contact with a flat surface as shown in Fig. 5.3(a), through flexion and ulnar deviation a contact path is created as shown in Fig. 5.3(b). The motion is done throughout 14 s, and due to a significant presence of noise, the contact position data is filtered using a rolling low pass filter with window size 100. The experiment was conducted with the finger on the surface facing $-y$, meaning the GT normal is $\mathbf{n}_{gt} = [0, -1, 0]$.

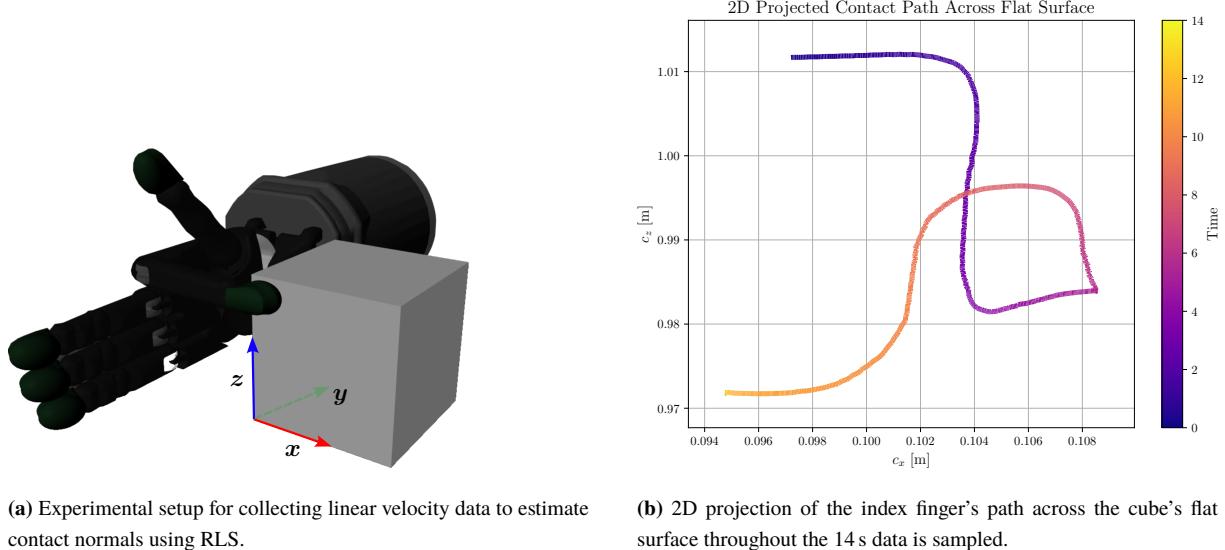


Fig. 5.3: Experimental setup and index finger's contact path when sampling contact data for normal estimation.

5.2.2 Skew Force Estimation

To test the performance of the DL model, four objects surfaces were used with known normals. These can be seen in Fig. 5.4 as a flat surface, an edge, a smooth surface and a corner.

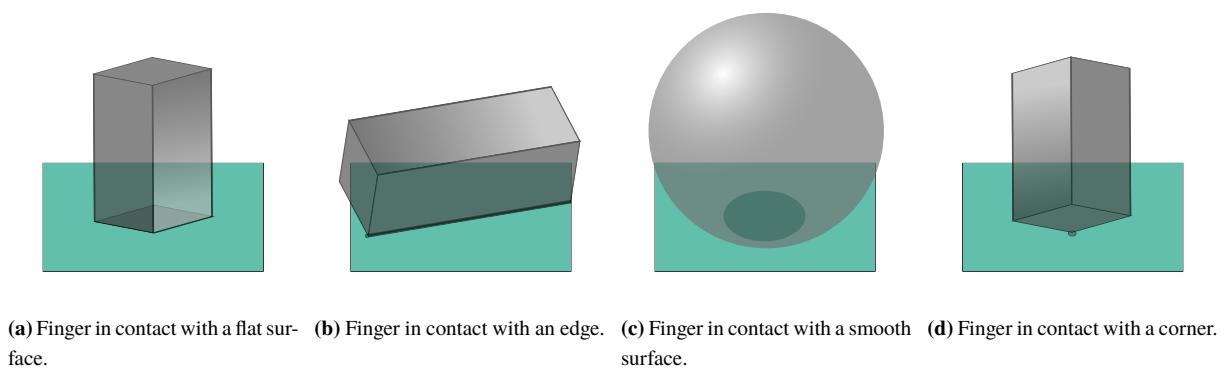


Fig. 5.4: The four surfaces used to test the performance of the DL model's ability to represent surfaces.

Within the simulation, the index finger is set to make contact with each surface, as shown in Fig. 5.5.

When contact is made the inputs and outputs of the DL model are recorded over 30 s, which with a sampling frequency of 100 Hz results in 3000 samples. As inputs are collected, the contact positions and forces are given



(a) Simulated index finger in contact with a flat surface. (b) Simulated index finger in contact with an edge. (c) Simulated index finger in contact with a smooth surface. (d) Simulated index finger in contact with a corner

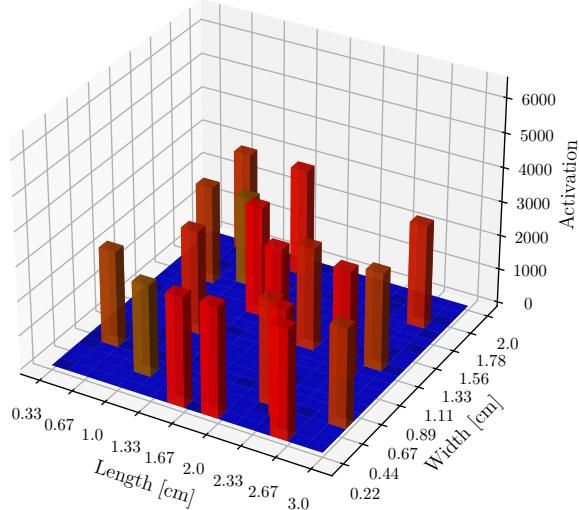
Fig. 5.5: The simulated Shadow Dexterous hand in contact with the four surfaces used to test the performance of the DL model's ability to represent surfaces. In each case, the contact is made by the index finger.

by Gazebo in $\{W\}$, which then is transformed into the contact frame $\{C\}$ using the grasping matrix \mathbf{G} , to ensure consistency. Due to contact data in Gazebo being prone to noise, an exponential decay filter is additionally applied.

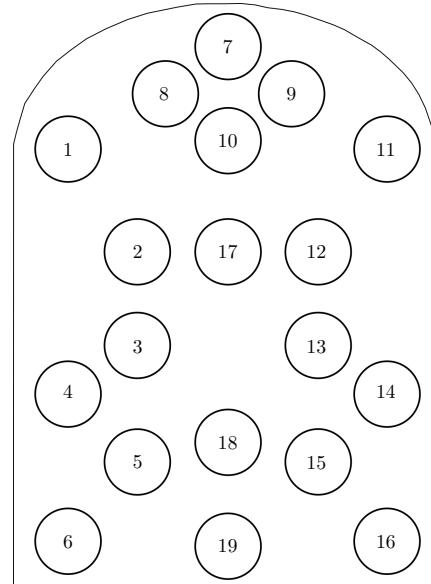
5.3 Results

After executing the DL model on all cases, the resulting simulated electrode activations were discovered to be infinite. Consequently, efforts were undertaken to address this issue. It was determined that the model does not include layer-wise normalization to establish limits on feature responses. The addition of this normalization procedure resulted in the network outputting values within the expected range. Fig. 5.6(a) shows a 3D plot of the electrode activations, while Fig. 5.6(b) shows a 2D projection of the finger tip with its labeled electrodes.

Simulated Electrode Measurements - Flat Surface



(a) 3D plot of electrode activations when the Shadow Dexterous hand's index finger makes contact with a flat surface.



(b) Map showing a 2D projection of the electrodes' positions and numbers.

Fig. 5.6: 3D plot of electrode activations and 2D electrode map with labels.

5.3.1 Contact Normals

Upon estimating the linear velocities Fig. 5.7 was produced, which shows the three velocity components. Due to the great presence of noise, a rolling low pass filter was applied with a window size of 100. As one would expect the v_y is of negligible magnitude, as shown here.

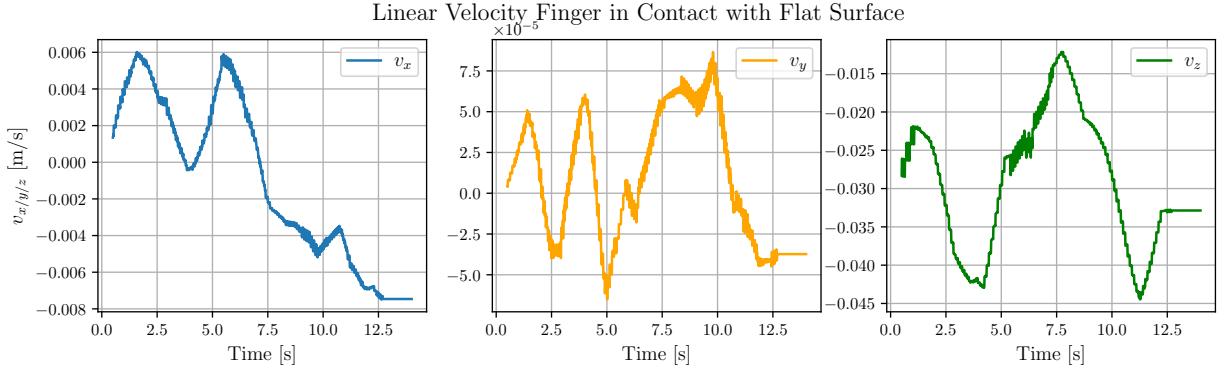


Fig. 5.7: Linear velocities of the contact points when the index finger moves across a flat surface.

Based on these the contact normals are estimated using RLS, which results in Fig. 5.8, which show great consistency and accuracy.

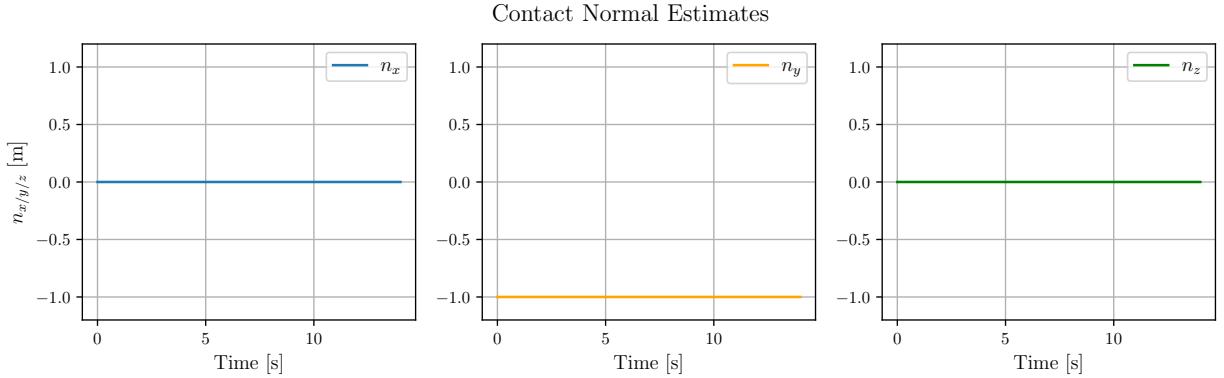


Fig. 5.8: The normal estimates across time as the experiment was executed.

5.3.2 Skew Forces

Fig. 5.9 illustrates the inputs and outputs of the DL model when the Shadow Dexterous hand's index finger makes contact with a flat surface. As seen here, the output of the model is independent of the input. To verify this behavior, data from the data set on which the model has been trained is applied and a similar result is found as seen in Fig. 5.10. The missing elements from this graph are value responses from the network of `inf`, which is a reference to the highest representable value by the system and is therefore not included.

Due to this independence of input, the electrodes, and by extension the DL model is not judged to be able to accurately simulate skew forces for a BioTac sensor in contact. The contact data from the remaining experiments can be found in Appendix B, which show a similar pattern.

5.3.3 Physics Engine Comparison

Gazebo's physics engine provides interpretations of tactile contacts for models containing a contact sensor model plugin. The data comes in the form of a `ContactState` which contain the contact points in $\{\mathbf{W}\}$, wrenches, depth and more. As a replacement for the lacking skew forces from the DL method, and a possible extension of the contact point and normal estimation, Gazebo's physics engine is considered a potential supplement to the methods presented.

Contact Forces and Electrode Activations - Flat Surface

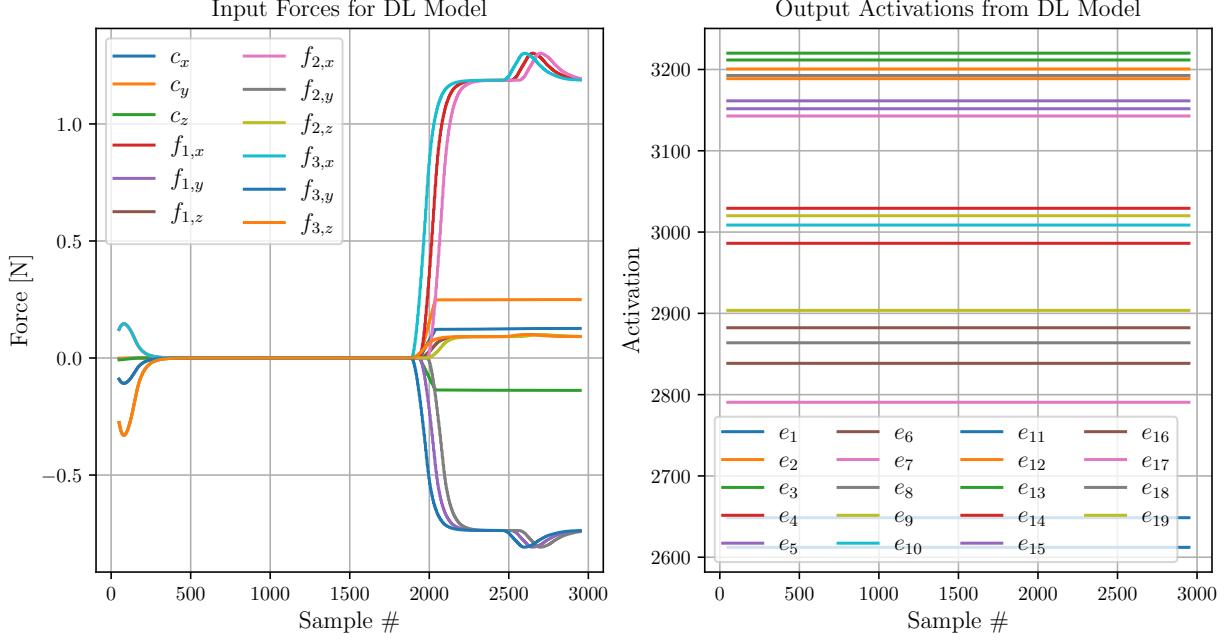


Fig. 5.9: The simulated tactile electrode activations the index finger is in contact with a flat surface.

Contact Forces and Electrode Activations - Train Data

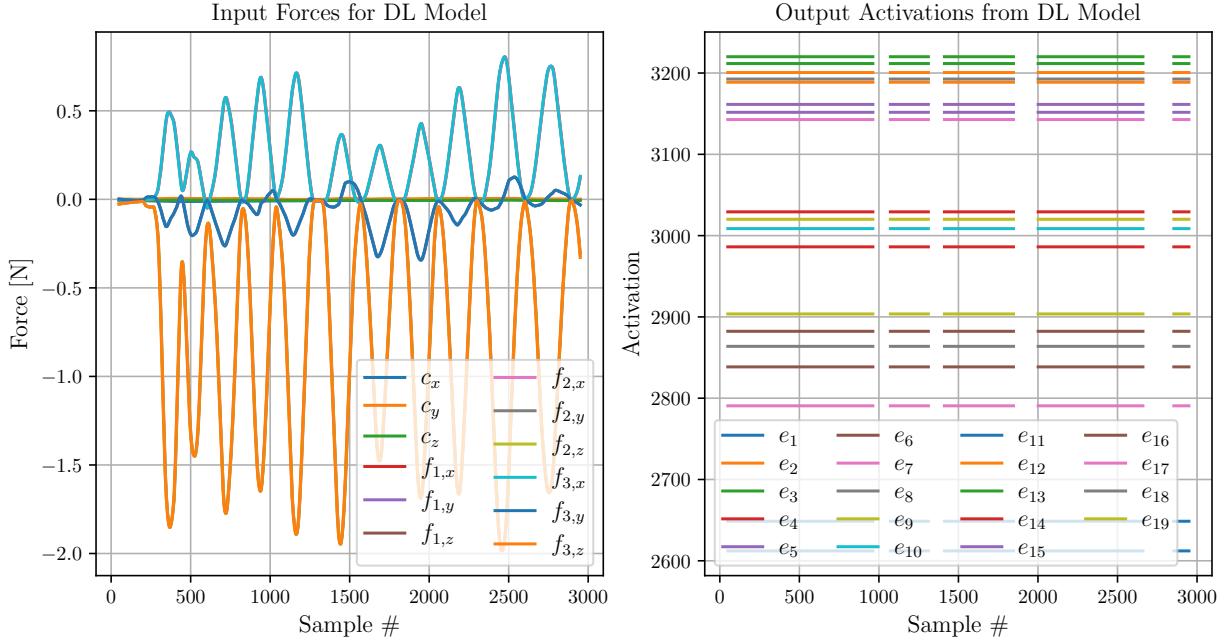
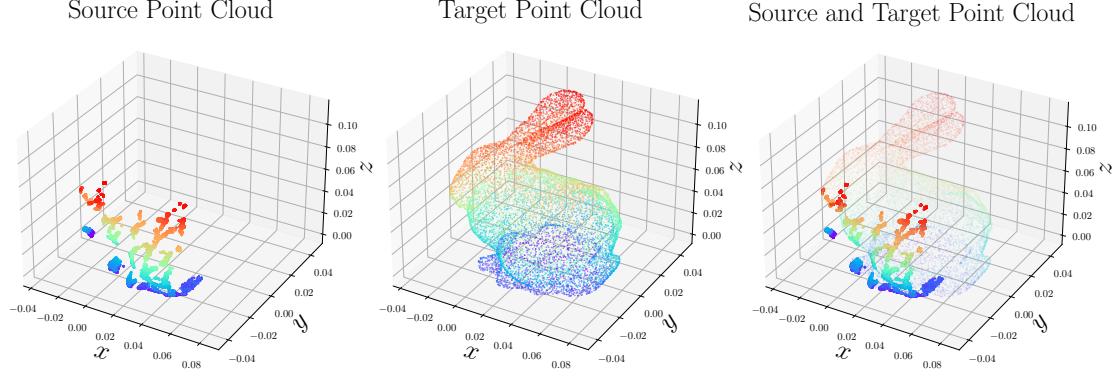


Fig. 5.10: The simulated tactile electrode activations when data the DL model was trained on is parsed.

To test the engine's ability to represent contact points, a 3D mesh [129] was sampled using the Shadow Dexterous hand, with the contact points being recorded as shown in Fig. 5.11. Here Fig. 5.11(a) shows the contact points sampled from moving the Shadow Dexterous fingers across the 3D mesh.

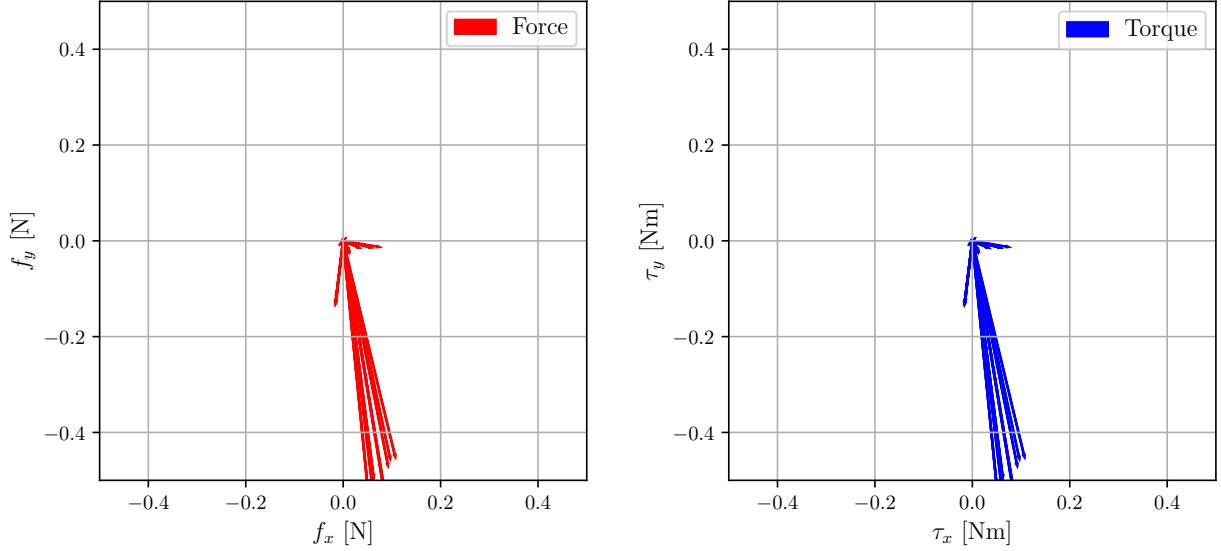
The contact forces and torques were found likewise found, shown in Fig. 5.12(a) and Fig. 5.12(b), for the Shadow Dexterous finger's being in contact with an edge.

The normals however posed a challenge, as the colliding meshes caused a misinterpretation by the physics engine to



(a) The source PC generated from simulated contact points. (b) The target PC generated by sampling the model mesh with 10 000 points. (c) The source PC overlaid the target PC, showing their fit.

Fig. 5.11: 3D plots showing the sampled source and target PCs along with a plot showing both of them overlaid.



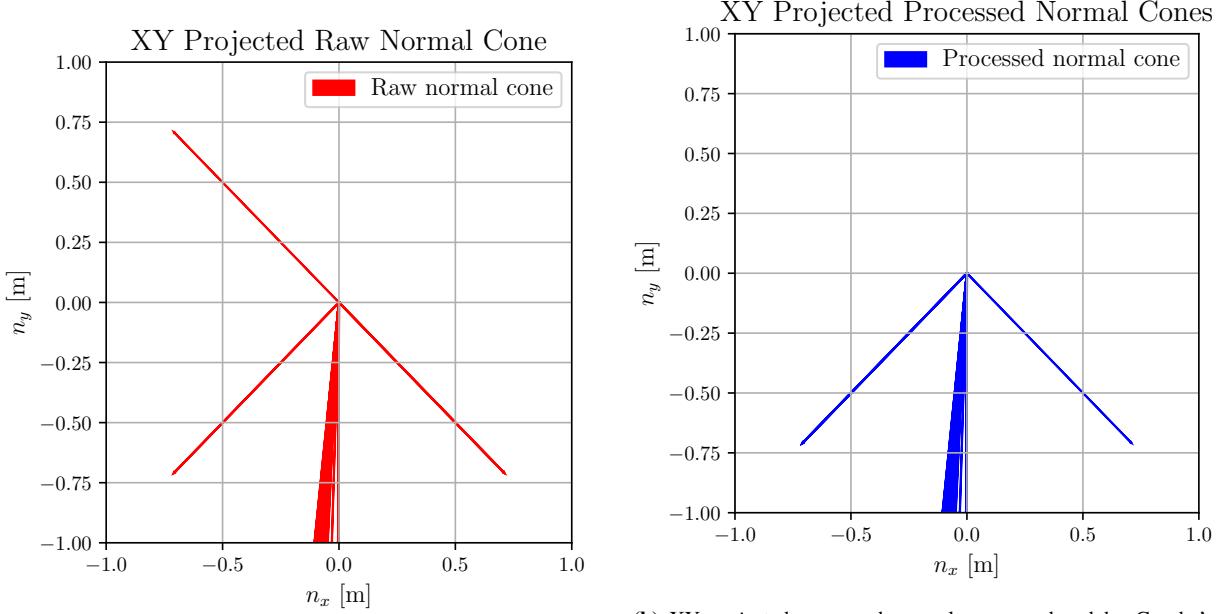
(a) XY projected force cone produced by Gazebo's physics engine when the Shadow Dexterous hand is in contact with an edge.

(b) XY projected torque cone produced by Gazebo's physics engine when the Shadow Dexterous hand is in contact with an edge.

Fig. 5.12: XY projected force and torque cones produced by Gazebo's physics engine when the Shadow Dexterous hand is in contact with an edge.

produce contact normals which were reflected 180°. This can be seen in Fig. 5.13(a). This was solved by sampling a set of contact normals over 10 time steps, reducing the sampling frequency to 10 Hz. These were then clustered using Euclidean clustering with an ϵ of 1 cm and 3 as the minimum number of samples. The centroid of each cluster was then used as the representative and the normal cones shown in Fig. 5.13(b) were achieved.

Finally, the angle errors θ_e of the contact normals were found for each of the four presented cases. This can be seen in Fig. 5.14



(a) XY projected raw normal cone produced by Gazebo's physics engine when the Shadow Dexterous hand is in contact with an edge.

(b) XY projected processed normal cone produced by Gazebo's physics engine when the Shadow Dexterous hand is in contact with an edge.

Fig. 5.13: XY projected raw and processed normal cones produced by Gazebo's physics engine when the Shadow Dexterous hand is in contact with an edge.

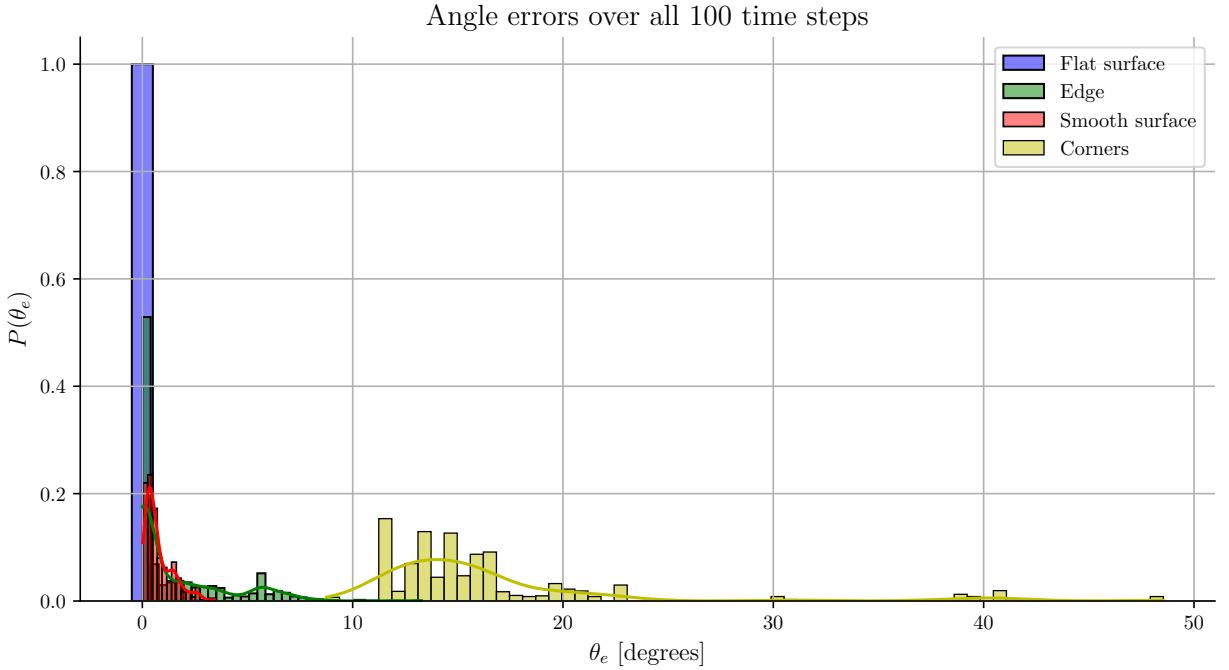


Fig. 5.14: Probability distributions of the angle error θ_e between simulated and GT contact normals.

5.4 Discussion & Conclusion

In this study, the performance of a DL model in estimating the tactile perception of a BioTac sensor when in contact with different objects is evaluated. The findings indicate that the DL model did not provide any useful information in this study. The lack of accuracy in simulating the electrode activations limits the usefulness of

the model in applications that require tactile force sensing. This finding highlights the importance of carefully evaluating the performance of DL models in specific applications and contexts, as they may not always provide the expected benefits. It was at the time of this project not possible to replicate the tactile information from the original paper [86].

Contact normals Instead, we used simulated contact points and the RLS method to estimate these values. However, the contact normals produced by Gazebo’s physics engine were acceptable and could be used to supplement our methods.

In conclusion, this study provides insights into the limitations of DL models in estimating tactile perception and highlights the importance of considering alternative methods, such as physics engine simulations, to supplement or replace DL models when necessary. Future studies could explore other DL architectures or combinations of different methods to improve the accuracy and usefulness of tactile perception estimation.

Chapter 6

Pose Estimation

6.1 Introduction

Here we write the introduction for problem 2.

In this chapter, the RCQP method will be presented along with its performance in solving the point cloud registration problem. The data produced in Chapter 5 is a point cloud of the form

$$\mathbf{X} = \begin{bmatrix} c_x & c_y & c_z & n_x & n_y & n_z \\ c_x & c_y & c_z & n_x & n_y & n_z \\ \vdots \\ c_x & c_y & c_z & n_x & n_y & n_z \end{bmatrix} \in \mathbb{R}^{M \times 6}, \quad (6.1)$$

where $\mathbf{c} = [c_x, c_y, c_z]$ is a contact point and $\mathbf{n} = [n_x, n_y, n_z]$ is the corresponding point's normals vector. Under the assumption of already knowing the object of interest, a GT point cloud \mathbf{Y} is also generated, which takes the same structure except the number of points being M . One row in the data matrix \mathbf{X} is referred to as \mathbf{x} .

The problem is thus to solve,

$$\mathbf{T}^* \arg \min_{\mathbf{T} \in \text{SE}(3)} \sum_{i=1}^M d_{P_i}(\mathbf{T}\mathbf{x}_i)^2 \quad (6.2)$$

where $\text{SE}(3)$ is the Special Euclidean group in 3D, $\mathbf{T}\mathbf{x}$ is the Euclidean transformation of the point \mathbf{x}_i and $d_{P_i}(\cdot)$ is the distance to the primitive P_i .

To solve this problem, correspondences must be found between the two point cloud matrices \mathbf{X} and \mathbf{Y} . Due to the points collected from

6.2 Method

The method chosen for this chapter is twofold: first outliers are rejected using GNC and RCQP for determining the optimal transformation \mathbf{T}^* . One of the common problems of PCR is the presence of outliers, exceeding 95 % is not uncommon [130]. Outlier removal is thus necessary, which is chosen in the form GNC. To find correspondences, a point cloud feature which utilizes the clusters of points produced by the fingertips. Thus the Fast Point Feature Histograms (FPFH) descriptor is chosen. Using these features descriptors d_t are found for the target data i.e. \mathbf{Y} and the source data \mathbf{X} i.e. d_s . Using these matching pairs of points are found using an exhaustive search and a similarity threshold of 0.01

the distance is the normalized Euclidean distance between the matching features.

The data is then organized in matrices \mathbf{X} and \mathbf{Y} for the source and target data, but organized such that each row corresponds

6.2.1 Graduated Non-Convexity

6.2.2 Relaxed Convex Quadratic Programming

6.3 Experimental Setup

In order to sample dat

Chapter 7

In-Hand Manipulation

7.1 Introduction

Here we write the introduction for problem 3.

7.2 Related Work

For history see hand book of robotics chapter 38, the first section.

Chapter 8

System Integration

8.1 Introduction

Here we write the introduction for the system integration.

Chapter 9

Discussion

Chapter 10

Conclusion

Bibliography

- [1] El Zaatari, Shirine et al. “Cobot programming for collaborative industrial tasks: An overview”. In: *Robotics and Autonomous Systems* 116 (June 2019), pp. 162–180. doi: [10.1016/j.robot.2019.03.003](https://doi.org/10.1016/j.robot.2019.03.003).
- [2] *What is computer vision?* <https://www.ibm.com/topics/computer-vision>. Accessed: 2022-11-02.
- [3] Zimmer, Henning. “Correspondence problems in computer vision”. PhD thesis. Jan. 2012.
- [4] LeCun, Yann, Bengio, Yoshua, and Hinton, Geoffrey. “Deep learning”. In: *Nature* 521.7553 (May 2015), pp. 436–444. issn: 1476-4687. doi: [10.1038/nature14539](https://doi.org/10.1038/nature14539). url: <https://doi.org/10.1038/nature14539>.
- [5] Monkman, Gareth J. et al. “Robot Grippers”. PhD thesis. Jan. 2004, pp. 5–6.
- [6] Mihelj, Matjaž et al. “Robotics Second Edition”. PhD thesis. Jan. 2019, p. 1.
- [7] Szeliski, Richard. *Computer vision: algorithms and applications*. Springer Nature, 2022, p. 284.
- [8] Quigley, Morgan et al. “ROS: an open-source Robot Operating System”. In: *ICRA Workshop on Open Source Software* 3 (Jan. 2009).
- [9] Publications, United Nations. “Inequality in Asia and the Pacific in the Era of the 2030 Agenda for Sustainable Development”. In: 7May 2018. url: <https://www.unescap.org/publications/inequality-asia-and-pacific-era-2030-agenda-sustainable-development>.
- [10] Wenwen, Shi et al. “Analysis and Control of Human Error”. In: *Procedia Engineering* 26 (Dec. 2011), pp. 2126–2132. doi: [10.1016/j.proeng.2011.11.2415](https://doi.org/10.1016/j.proeng.2011.11.2415).
- [11] Galin, Rinat et al. “Cobots and the benefits of their implementation in intelligent manufacturing”. In: *IOP Conference Series: Materials Science and Engineering* 862 (May 2020), p. 032075. doi: [10.1088/1757-899X/862/3/032075](https://doi.org/10.1088/1757-899X/862/3/032075).
- [12] Raptopoulos, Fredy, Koskinopoulou, Maria, and Maniadakis, Michail. “Robotic Pick-and-Toss Facilitates Urban Waste Sorting”. In: *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. 2020, pp. 1149–1154. doi: [10.1109/CASE48305.2020.9216746](https://doi.org/10.1109/CASE48305.2020.9216746).
- [13] Talpur, Mir Sajjad Hussain and Shaikh, Murtaza Hussain. *Automation of Mobile Pick and Place Robotic System for Small Food Industry*. 2012. doi: [10.48550/ARXIV.1203.4475](https://arxiv.org/abs/1203.4475). url: <https://arxiv.org/abs/1203.4475>.
- [14] Yamanaka, Yuta et al. “Development of a Food Handling Soft Robot Hand Considering a High-speed Pick-and-place Task”. In: *2020 IEEE/SICE International Symposium on System Integration (SII)*. 2020, pp. 87–92. doi: [10.1109/SII46433.2020.9026282](https://doi.org/10.1109/SII46433.2020.9026282).
- [15] Lee, Sukhan and Lee, Yeonho. “Real-Time Industrial Bin-Picking with a Hybrid Deep Learning-Engineering Approach”. In: *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*. 2020, pp. 584–588. doi: [10.1109/BigComp48618.2020.00015](https://doi.org/10.1109/BigComp48618.2020.00015).
- [16] Mnyusiwalla, Hussein et al. “A Bin-Picking Benchmark for Systematic Evaluation of Robotic Pick-and-Place Systems”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1389–1396. doi: [10.1109/LRA.2020.2965076](https://doi.org/10.1109/LRA.2020.2965076).
- [17] Wong, Ching-Chang et al. “Generic Development of Bin Pick-and-Place System Based on Robot Operating System”. In: *IEEE Access* 10 (2022), pp. 65257–65270. doi: [10.1109/ACCESS.2022.3182114](https://doi.org/10.1109/ACCESS.2022.3182114).

- [18] He, Zaixing et al. “6D Pose Estimation of Objects: Recent Technologies and Challenges”. In: *Applied Sciences* 11.1 (2021). ISSN: 2076-3417. doi: 10.3390/app11010228. URL: <https://www.mdpi.com/2076-3417/11/1/228>.
- [19] .., Taryudi and Wang, Ming-Shyan. “3D object pose estimation using stereo vision for object manipulation system”. In: May 2017, pp. 1532–1535. doi: 10.1109/ICASI.2017.7988217.
- [20] Oh, Jong-Kyu, Lee, Sukhan, and Lee, Chan-Ho. “Stereo vision based automation for a bin-picking solution”. In: *International Journal of Control, Automation and Systems* 10.2 (Apr. 2012), pp. 362–373. ISSN: 2005-4092. doi: 10.1007/s12555-012-0216-9. URL: <https://doi.org/10.1007/s12555-012-0216-9>.
- [21] Abdelaal, Mahmoud et al. “Uncalibrated stereo vision with deep learning for 6-DOF pose estimation for a robot arm system”. In: *Robotics and Autonomous Systems* 145 (2021), p. 103847. ISSN: 0921-8890. doi: <https://doi.org/10.1016/j.robot.2021.103847>. URL: <https://www.sciencedirect.com/science/article/pii/S0921889021001329>.
- [22] Nakano, Yoshihiro. “Stereo Vision Based Single-Shot 6D Object Pose Estimation for Bin-Picking by a Robot Manipulator”. In: *CoRR* abs/2005.13759 (2020). arXiv: 2005.13759. URL: <https://arxiv.org/abs/2005.13759>.
- [23] Kozák, Viktor et al. “Data-Driven Object Pose Estimation in a Practical Bin-Picking Application”. In: *Sensors* 21.18 (2021). ISSN: 1424-8220. doi: 10.3390/s21186093. URL: <https://www.mdpi.com/1424-8220/21/18/6093>.
- [24] Xu, Chi et al. “6DoF Pose Estimation of Transparent Object from a Single RGB-D Image”. In: *Sensors* 20.23 (2020). ISSN: 1424-8220. doi: 10.3390/s20236790. URL: <https://www.mdpi.com/1424-8220/20/23/6790>.
- [25] Straszheim, Troy et al. *Shadow Robot*. URL: <https://www.shadowrobot.com/dexterous-hand-series/>. (accessed: 29.07.2022).
- [26] Koenig, N. and Howard, A. “Design and use paradigms for Gazebo, an open-source multi-robot simulator”. In: 3 (2004), 2149–2154 vol.3. doi: 10.1109/IROS.2004.1389727.
- [27] docker. *What is a container*. URL: <https://www.docker.com/resources/what-container/>.
- [28] Robotics, Shadow. *Shadow Robotics*. URL: <https://www.shadowrobot.com/>.
- [29] Robotics, Shadow. *The Shadow Robot Company*. URL: <https://github.com/shadow-robot>.
- [30] Dimensions. https://shadow-robot-company-dexterous-hand.readthedocs-hosted.com/en/latest/user_guide/md_dimensions.html. Accessed: 2023-04-26.
- [31] Rahman, Akhlaqur and Al-Jumaily, Adel. “Design and Development of a Bilateral Therapeutic Hand Device for Stroke Rehabilitation”. In: *International Journal of Advanced Robotic Systems* 10.12 (2013), p. 405. doi: 10.5772/56809. eprint: <https://doi.org/10.5772/56809>. URL: <https://doi.org/10.5772/56809>.
- [32] Bones Of The Hand. <https://freesvg.org/bones-of-the-hand>. Accessed: 2023-04-26.
- [33] Robotics, Shadow. Gazebo. URL: %5Curl%7Bhttps://shadow-robot-company-dexterous-hand.readthedocs-hosted.com/en/latest/user_guide/sim_gazebo.html%7D.
- [34] First time users. https://shadow-robot-company-dexterous-hand.readthedocs-hosted.com/en/latest/user_guide/sd_first_time_users.html. Accessed: 2023-04-26.
- [35] Straszheim, Troy et al. *Conceptual overview of ROS catkin*. URL: http://wiki.ros.org/catkin/conceptual_overview.
- [36] Beek, Bas van et al. *About Us*. URL: %5Curl%7B<https://numpy.org/%7D>.

- [37] team, OpenCV. *OpenCV*. URL: %5Curl%7Bhttps://opencv.org/%7D.
- [38] Carroll, Michael. *dynamic_reconfigure*. URL: %5Curl%7Bhttp://wiki.ros.org/dynamic_reconfigure%7D.
- [39] Coleman, David et al. *Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study*. 2014. arXiv: 1404.3785 [cs.RO].
- [40] Sucan, Ioan A., Moll, Mark, and Kavraki, Lydia E. “The Open Motion Planning Library”. In: *IEEE Robotics & Automation Magazine* 19.4 (Dec. 2012). <https://ompl.kavrakilab.org>, pp. 72–82. doi: 10.1109/MRA.2012.2205651.
- [41] Chitta, Sachin et al. “ros_control: A generic and simple control framework for ROS”. In: *The Journal of Open Source Software* (2017). doi: 10.21105/joss.00456. URL: <http://www.theoj.org/joss-papers/joss.00456/10.21105.joss.00456.pdf>.
- [42] Robotics, Shadow. *Control Description*. URL: %5Curl%7Bhttps://shadow-robot-company-dexterous-hand.readthedocs-hosted.com/en/latest/user_guide/ed_control_description.html%7D.
- [43] Robotics, Shadow. *Firmware*. URL: %5Curl%7Bhttps://shadow-robot-company-dexterous-hand.readthedocs-hosted.com/en/latest/user_guide/sd_firmware.html%7D.
- [44] Robotics, Shadow. *Controlling the Hand*. URL: %5Curl%7Bhttps://shadow-robot-company-dexterous-hand.readthedocs-hosted.com/en/latest/user_guide/sd_controlling_hand.html%7D.
- [45] Pérez-González, Antonio et al. “A modified elastic foundation contact model for application in 3D models of the prosthetic knee”. In: *Medical Engineering & Physics* 30.3 (2008), pp. 387–398. ISSN: 1350-4533. doi: <https://doi.org/10.1016/j.medengphy.2007.04.001>. URL: <https://www.sciencedirect.com/science/article/pii/S1350453307000616>.
- [46] Hertz, H. “On the Contact of Rigid Elastic Solids and on Hardness”. In: *Ch 6: Assorted Papers* (1882).
- [47] Bruno Siciliano, Oussama Khatib, ed. *Springer Handbook of Robotics*. Springer Berlin, Heidelberg, 2016.
- [48] Xydas, Nicholas and Kao, Imin. “Modeling of Contact Mechanics and Friction Limit Surfaces for Soft Fingers in Robotics, with Experimental Results”. In: *The International Journal of Robotics Research* 18.9 (1999), pp. 941–950. doi: 10.1177/02783649922066673. eprint: <https://doi.org/10.1177/02783649922066673>. URL: <https://doi.org/10.1177/02783649922066673>.
- [49] Yuvaraj, S., Malayalamurthi, R., and Raja, K. Venkatesh. “The haptic and perceptual characteristics of an anthropomorphic curved soft finger structure”. In: *Curved and Layered Structures* 6.1 (2019), pp. 161–168. doi: [doi:10.1515/cls-2019-0013](https://doi.org/10.1515/cls-2019-0013). URL: <https://doi.org/10.1515/cls-2019-0013>.
- [50] Howe, R.D., Kao, I., and Cutkosky, M.R. “The sliding of robot fingers under combined torsion and shear loading”. In: *Proceedings. 1988 IEEE International Conference on Robotics and Automation*. 1988, 103–105 vol.1. doi: 10.1109/ROBOT.1988.12032.
- [51] Flugge, Wilhelm. *Viscoelasticity*. Blaisdell Publishing Company, Waltham, MA, 1967.
- [52] Maxwell, James Clerk. *On the dynamical theory of gases*. Royal Society, 1January 1867.
- [53] Fung, Yuan-Cheng. “Mechanical properties and active remodeling of blood vessels”. In: *Biomechanics*. Springer, 1993.
- [54] Tiezzi, Paolo and Kao, Imin. “Modeling of Viscoelastic Contacts and Evolution of Limit Surface for Robotic Contact Interface”. In: *Robotics, IEEE Transactions on* 23 (May 2007), pp. 206–217. doi: 10.1109/TRO.2006.889494.

- [55] Tiezzi, P. and Kao, I. “Characteristics of contact and limit surface for viscoelastic fingers”. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. 2006, pp. 1365–1370. doi: [10.1109/ROBOT.2006.1641899](https://doi.org/10.1109/ROBOT.2006.1641899).
- [56] Tiezzi, Paolo, Kao, Imin, and Vassura, G. “Effect of layer compliance on frictional behavior of soft robotic fingers”. In: *Advanced Robotics* 21 (Oct. 2007), pp. 1653–1670. doi: [10.1163/156855307782227390](https://doi.org/10.1163/156855307782227390).
- [57] Kalker, J. J. “On the Contact Problem in Elastostatics”. In: *Unilateral Problems in Structural Analysis*. Ed. by Del Piero, Gianpietro and Maceri, Franco. Vienna: Springer Vienna, 1985, pp. 81–85. ISBN: 978-3-7091-2632-.
- [58] Kozhevnikov, I.F. et al. “A new algorithm for computing the indentation of a rigid body of arbitrary shape on a viscoelastic half-space”. In: *International Journal of Mechanical Sciences* 50.7 (2008), pp. 1194–1202. ISSN: 0020-7403. doi: <https://doi.org/10.1016/j.ijmecsci.2008.04.003>. URL: <https://www.sciencedirect.com/science/article/pii/S0020740308000623>.
- [59] Marmo, Francesco and Rosati, Luciano. “A General Approach to the Solution of Boussinesq’s Problem for Polynomial Pressures Acting over Polygonal Domains”. In: *Journal of Elasticity* 122.1 (Jan. 2016), pp. 75–112. ISSN: 1573-2681. doi: [10.1007/s10659-015-9534-5](https://doi.org/10.1007/s10659-015-9534-5). URL: <https://doi.org/10.1007/s10659-015-9534-5>.
- [60] Li, Junshan and Berger, Edward J. “A Boussinesq–Cerruti Solution Set for Constant and Linear Distribution of Normal and Tangential Load over a Triangular Area”. In: *Journal of elasticity and the physical science of solids* 63.2 (May 2001), pp. 137–151. ISSN: 1573-2681. doi: [10.1023/A:1014013425423](https://doi.org/10.1023/A:1014013425423). URL: <https://doi.org/10.1023/A:1014013425423>.
- [61] Wasko, Wojciech et al. “Contact Modelling and Tactile Data Processing for Robot Skins”. In: *Sensors* 19.4 (2019). ISSN: 1424-8220. doi: [10.3390/s19040814](https://doi.org/10.3390/s19040814). URL: <https://www.mdpi.com/1424-8220/19/4/814>.
- [62] Slaughter, William S. *The Linearized Theory of Elasticity*. Springer, 2012.
- [63] Hills, D., Nowell, D., and Barber, James. “KL Johnson and contact mechanics”. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 231 (Feb. 2016). doi: [10.1177/0954406216634121](https://doi.org/10.1177/0954406216634121).
- [64] Lee, S.Y., Kuo, Y.H., and Lin, F.Y. “Stability of a Timoshenko beam resting on a Winkler elastic foundation”. In: *Journal of Sound and Vibration* 153.2 (1992), pp. 193–202. ISSN: 0022-460X. doi: [https://doi.org/10.1016/S0022-460X\(05\)80001-X](https://doi.org/10.1016/S0022-460X(05)80001-X). URL: <https://www.sciencedirect.com/science/article/pii/S0022460X0580001X>.
- [65] Eisenberger, M. and Clastornik, J. “Vibrations and buckling of a beam on a variable winkle elastic foundation”. In: *Journal of Sound and Vibration* 115.2 (1987), pp. 233–241. ISSN: 0022-460X. doi: [https://doi.org/10.1016/0022-460X\(87\)90469-X](https://doi.org/10.1016/0022-460X(87)90469-X). URL: <https://www.sciencedirect.com/science/article/pii/0022460X8790469X>.
- [66] Fregly, Benjamin J., Bei, Yanhong, and Sylvester, Mark E. “Experimental evaluation of an elastic foundation model to predict contact pressures in knee replacements”. In: *Journal of Biomechanics* 36.11 (2003), pp. 1659–1668. ISSN: 0021-9290. doi: [https://doi.org/10.1016/S0021-9290\(03\)00176-3](https://doi.org/10.1016/S0021-9290(03)00176-3). URL: <https://www.sciencedirect.com/science/article/pii/S0021929003001763>.
- [67] Kao, Imin and Cutkosky, Mark R. “Quasistatic Manipulation with Compliance and Sliding”. In: *The International Journal of Robotics Research* 11.1 (1992), pp. 20–40. doi: [10.1177/027836499201100102](https://doi.org/10.1177/027836499201100102). eprint: <https://doi.org/10.1177/027836499201100102>. URL: <https://doi.org/10.1177/027836499201100102>.

- [68] Howe, Robert D. and Cutkosky, Mark R. “Practical Force-Motion Models for Sliding Manipulation”. In: *The International Journal of Robotics Research* 15.6 (1996), pp. 557–572. doi: 10.1177/027836499601500603. eprint: <https://doi.org/10.1177/027836499601500603>. URL: <https://doi.org/10.1177/027836499601500603>.
- [69] Kao, Imin and Yang, Fuqian. “Stiffness and Contact Mechanics for Soft Fingers in Grasping and Manipulation”. In: *Robotics and Automation, IEEE Transactions on* 20 (Mar. 2004), pp. 132–135. doi: 10.1109/TRA.2003.820868.
- [70] Ciocarlie, Matei, Lackner, Claire, and Allen, Peter. “Soft Finger Model with Adaptive Contact Geometry for Grasping and Manipulation Tasks”. In: *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*. 2007, pp. 219–224. doi: 10.1109/WHC.2007.103.
- [71] Sabat, Lovely and Kundu, Chinmay Kumar. “History of Finite Element Method: A Review”. In: *Recent Developments in Sustainable Infrastructure*. Ed. by Das, Bibhuti Bhushan et al. Singapore: Springer Singapore, 2021, pp. 395–404. ISBN: 978-981-15-4577-1.
- [72] Klocke, Fritz et al. “Examples of FEM application in manufacturing technology”. In: *Journal of Materials Processing Technology* 120.1 (2002), pp. 450–457. ISSN: 0924-0136. doi: [https://doi.org/10.1016/S0924-0136\(01\)01210-9](https://doi.org/10.1016/S0924-0136(01)01210-9). URL: <https://www.sciencedirect.com/science/article/pii/S0924013601012109>.
- [73] Telliskivi, T and Olofsson, U. “Contact mechanics analysis of measured wheel-rail profiles using the finite element method”. In: *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 215.2 (2001), pp. 65–72. doi: 10.1243/0954409011531404. eprint: <https://doi.org/10.1243/0954409011531404>. URL: <https://doi.org/10.1243/0954409011531404>.
- [74] Põdra, Priit and Andersson, Sören. “Simulating sliding wear with finite element method”. In: *Tribology International* 32.2 (1999), pp. 71–81. ISSN: 0301-679X. doi: [https://doi.org/10.1016/S0301-679X\(99\)00012-2](https://doi.org/10.1016/S0301-679X(99)00012-2). URL: <https://www.sciencedirect.com/science/article/pii/S0301679X99000122>.
- [75] Pantuso, Daniel, Bathe, Klaus-Jürgen, and Bouzinov, Pavel A. “A finite element procedure for the analysis of thermo-mechanical solids in contact”. In: *Computers & Structures* 75.6 (2000), pp. 551–573. ISSN: 0045-7949. doi: [https://doi.org/10.1016/S0045-7949\(99\)00212-6](https://doi.org/10.1016/S0045-7949(99)00212-6). URL: <https://www.sciencedirect.com/science/article/pii/S0045794999002126>.
- [76] Liang, Xiaodong, Ali, Mohammad Zawad, and Zhang, Huaguang. “Induction Motors Fault Diagnosis Using Finite Element Method: A Review”. In: *IEEE Transactions on Industry Applications* 56.2 (2020), pp. 1205–1217. doi: 10.1109/TIA.2019.2958908.
- [77] Ye, Zhiqiu et al. “Soft Robot Skin With Conformal Adaptability for On-Body Tactile Perception of Collaborative Robots”. In: *IEEE Robotics and Automation Letters* (Mar. 2022), pp. 1–1. doi: 10.1109/LRA.2022.3155225.
- [78] Lu, Guan, Fu, Shiwen, and Xu, Yiming. “Design and Experimental Research of Robot Finger Sliding Tactile Sensor Based on FBG”. In: *Sensors* 22.21 (2022). ISSN: 1424-8220. doi: 10.3390/s22218390. URL: <https://www.mdpi.com/1424-8220/22/21/8390>.
- [79] Ciocarlie, M., Miller, A., and Allen, P. “Grasp analysis using deformable fingers”. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2005, pp. 4122–4128. doi: 10.1109/IROS.2005.1545525.
- [80] Narang, Yashraj S. et al. “Sim-to-Real for Robotic Tactile Sensing via Physics-Based Simulation and Learned Latent Projections”. In: *CoRR* abs/2103.16747 (2021). arXiv: 2103.16747. URL: <https://arxiv.org/abs/2103.16747>.

- [81] Narang, Yashraj S. et al. “Interpreting and Predicting Tactile Signals via a Physics-Based and Data-Driven Framework”. In: *CoRR* abs/2006.03777 (2020). arXiv: 2006.03777. URL: <https://arxiv.org/abs/2006.03777>.
- [82] Sferrazza, Carmelo et al. “Ground Truth Force Distribution for Learning-Based Tactile Sensing: A Finite Element Approach”. In: *IEEE Access* PP (Nov. 2019), pp. 1–1. doi: 10.1109/ACCESS.2019.2956882.
- [83] Calandra, Roberto et al. “More Than a Feeling: Learning to Grasp and Regrasp using Vision and Touch”. In: *CoRR* abs/1805.11085 (2018). arXiv: 1805.11085. URL: <http://arxiv.org/abs/1805.11085>.
- [84] Spiers, Adam et al. “Single-Grasp Object Classification and Feature Extraction with Simple Robot Hands and Tactile Sensors”. In: *IEEE Transactions on Haptics* 9 (Jan. 2016), pp. 60–71. doi: 10.1109/TOH.2016.2521378.
- [85] Higuera, Carolina, Boots, Byron, and Mukadam, Mustafa. *Learning to Read Braille: Bridging the Tactile Reality Gap with Diffusion Models*. 2023. arXiv: 2304.01182 [cs.R0].
- [86] Ruppel, Philipp et al. “Simulation of the SynTouch BioTac Sensor”. In: (2019). Ed. by Strand, Marcus et al., pp. 374–387.
- [87] Huang, Xiaoshui et al. *A comprehensive survey on point cloud registration*. 2021. arXiv: 2103.02690 [cs.CV].
- [88] Oberweger, Markus, Wohlhart, Paul, and Lepetit, Vincent. *Hands Deep in Deep Learning for Hand Pose Estimation*. 2016. arXiv: 1502.06807 [cs.CV].
- [89] Toshev, Alexander and Szegedy, Christian. “DeepPose: Human Pose Estimation via Deep Neural Networks”. In: *Computer Vision and Pattern Recognition*. 2014.
- [90] Mathis, Alexander et al. “DeepLabCut: markerless pose estimation of user-defined body parts with deep learning”. In: *Nature Neuroscience* 21.9 (Sept. 2018), pp. 1281–1289. ISSN: 1546-1726. doi: 10.1038/s41593-018-0209-y. URL: <https://doi.org/10.1038/s41593-018-0209-y>.
- [91] Huang, Xiaoshui et al. “A Systematic Approach for Cross-Source Point Cloud Registration by Preserving Macro and Micro Structures”. In: *IEEE Transactions on Image Processing* 26.7 (2017), pp. 3261–3276. doi: 10.1109/TIP.2017.2695888.
- [92] Huang, Xiaoshui et al. “A Coarse-to-Fine Algorithm for Matching and Registration in 3D Cross-Source Point Clouds”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 28.10 (2018), pp. 2965–2977. doi: 10.1109/TCSVT.2017.2730232.
- [93] Yang, Zhenpei et al. “Extreme Relative Pose Estimation for RGB-D Scans via Scene Completion”. In: June 2019, pp. 4526–4535. doi: 10.1109/CVPR.2019.00466.
- [94] Wang, Lingjing et al. *Non-Rigid Point Set Registration Networks*. 2019. arXiv: 1904.01428 [cs.GR].
- [95] Huang, Xiaoshui, Mei, Guofeng, and Zhang, Jian. *Feature-metric Registration: A Fast Semi-supervised Approach for Robust Point Cloud Registration without Correspondences*. 2020. arXiv: 2005.01014 [cs.CV].
- [96] Zeng, Andy et al. *3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions*. 2017. arXiv: 1603.08182 [cs.CV].
- [97] Deng, Haowen, Birdal, Tolga, and Ilic, Slobodan. *PPFNet: Global Context Aware Local Features for Robust 3D Point Matching*. 2018. arXiv: 1802.02669 [cs.CV].
- [98] Gojcic, Zan et al. *The Perfect Match: 3D Point Cloud Matching with Smoothed Densities*. 2019. arXiv: 1811.06879 [cs.CV].
- [99] Besl, Paul J. and McKay, Neil D. “A Method for Registration of 3-D Shapes”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (1992), pp. 239–256.

- [100] Chen, Yang and Medioni, Gérard. “Object modelling by registration of multiple range images”. In: *Image and Vision Computing* 10.3 (1992). Range Image Understanding, pp. 145–155. issn: 0262-8856. doi: [https://doi.org/10.1016/0262-8856\(92\)90066-C](https://doi.org/10.1016/0262-8856(92)90066-C). url: <https://www.sciencedirect.com/science/article/pii/026288569290066C>.
- [101] Segal, Aleksandr, Hähnel, Dirk, and Thrun, Sebastian. “Generalized-ICP”. In: June 2009. doi: [10.15607/RSS.2009.V.021](https://doi.org/10.15607/RSS.2009.V.021).
- [102] Shi, Xiaojing, Liu, Tao, and Han, Xie. “Improved Iterative Closest Point(ICP) 3D point cloud registration algorithm based on point cloud filtering and adaptive fireworks for coarse registration”. In: *International Journal of Remote Sensing* 41.8 (2020), pp. 3197–3220. doi: [10.1080/01431161.2019.1701211](https://doi.org/10.1080/01431161.2019.1701211). eprint: <https://doi.org/10.1080/01431161.2019.1701211>. url: <https://doi.org/10.1080/01431161.2019.1701211>.
- [103] Zhu, Hao et al. “A Review of Point Set Registration: From Pairwise Registration to Groupwise Registration”. en. In: *Sensors (Basel)* 19.5 (Mar. 2019).
- [104] Livi, Lorenzo and Rizzi, Antonello. “The Graph Matching Problem”. In: *Pattern Anal. Appl.* 16.3 (Aug. 2013), pp. 253–283. issn: 1433-7541. doi: [10.1007/s10044-012-0284-8](https://doi.org/10.1007/s10044-012-0284-8). url: <https://doi.org/10.1007/s10044-012-0284-8>.
- [105] Zhou, Feng and De la Torre, Fernando. “Factorized Graph Matching”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.9 (2016), pp. 1774–1789. doi: [10.1109/TPAMI.2015.2501802](https://doi.org/10.1109/TPAMI.2015.2501802).
- [106] Leordeanu, Marius and Hebert, Martial. “A spectral technique for correspondence problems using pairwise constraints”. In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 2* (2005), 1482–1489 Vol. 2.
- [107] Zass, Ron and Shashua, Amnon. “Probabilistic graph and hypergraph matching”. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 2008, pp. 1–8. doi: [10.1109/CVPR.2008.4587500](https://doi.org/10.1109/CVPR.2008.4587500).
- [108] Duchenne, Olivier et al. “A Tensor-Based Algorithm for High-Order Graph Matching”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.12 (2011), pp. 2383–2395. doi: [10.1109/TPAMI.2011.110](https://doi.org/10.1109/TPAMI.2011.110).
- [109] Zhu, Hu et al. “Elastic Net Constraint-Based Tensor Model for High-Order Graph Matching”. In: *IEEE Transactions on Cybernetics* 51.8 (2021), pp. 4062–4074. doi: [10.1109/TCYB.2019.2936176](https://doi.org/10.1109/TCYB.2019.2936176).
- [110] Müller, Meinard. *Information Retrieval for Music and Motion*. Jan. 2007. isbn: 978-3-540-74047-6. doi: [10.1007/978-3-540-74048-3](https://doi.org/10.1007/978-3-540-74048-3).
- [111] Fan, Jingfan et al. “Convex hull indexed Gaussian mixture model (CH-GMM) for 3D point set registration”. In: *Pattern Recognition* 59 (2016). Compositional Models and Structured Learning for Visual Recognition, pp. 126–141. issn: 0031-3203. doi: <https://doi.org/10.1016/j.patcog.2016.02.023>. url: <https://www.sciencedirect.com/science/article/pii/S0031320316000947>.
- [112] Evangelidis, Georgios D. et al. “A Generative Model for the Joint Registration of Multiple Point Sets”. In: *Computer Vision – ECCV 2014*. Ed. by Fleet, David et al. Cham: Springer International Publishing, 2014, pp. 109–122. isbn: 978-3-319-10584-0.
- [113] Yuan, Wentao et al. *DeepGMR: Learning Latent Gaussian Mixture Models for Registration*. 2020. arXiv: [2008.09088 \[cs.CV\]](https://arxiv.org/abs/2008.09088).
- [114] Yang, Heng, Shi, Jingnan, and Carlone, Luca. “TEASER: Fast and Certifiable Point Cloud Registration”. In: *IEEE Transactions on Robotics* 37.2 (2021), pp. 314–333. doi: [10.1109/TRO.2020.3033695](https://doi.org/10.1109/TRO.2020.3033695).
- [115] Sun, Lei. *IRON: Invariant-based Highly Robust Point Cloud Registration*. 2021. arXiv: [2103.04357 \[cs.CV\]](https://arxiv.org/abs/2103.04357).

- [116] Briales, Jesus and Gonzalez-Jimenez, Javier. “Convex Global 3D Registration With Lagrangian Duality”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.
- [117] Yang, Heng et al. “Graduated Non-Convexity for Robust Spatial Perception: From Non-Minimal Solvers to Global Outlier Rejection”. In: *IEEE Robotics and Automation Letters* 5.2 (Apr. 2020), pp. 1127–1134. doi: [10.1109/lra.2020.2965893](https://doi.org/10.1109/lra.2020.2965893). URL: <https://doi.org/10.1109/lra.2020.2965893>.
- [118] Salisbury, J. Kenneth and Craig, John J. “Articulated Hands: Force Control and Kinematic Issues”. In: *The International Journal of Robotics Research* 1.1 (1982), pp. 4–17. doi: [10.1177/027836498200100102](https://doi.org/10.1177/027836498200100102). eprint: <https://doi.org/10.1177/027836498200100102>. URL: <https://doi.org/10.1177/027836498200100102>.
- [119] Jonker, Ben, Waiboer, Rob, and Aarts, Ronald. “Modelling of joint friction in robotic manipulators with gear transmissions”. In: Jan. 2007, pp. 221–243. ISBN: 10-1-4020-5683-4.
- [120] Lynch, Kevin M and Park, Frank C. *Modern Robotics: Mechanics, Planning, and Control*. English (US). Cambridge Univeristy Press, 2017. ISBN: 978-1107156302.
- [121] Ghaednia, Hamed et al. “Contact Mechanics”. In: Nov. 2013, pp. 93–140. ISBN: 978-1-4614-1944-0. doi: [10.1007/978-1-4614-1945-7_3](https://doi.org/10.1007/978-1-4614-1945-7_3).
- [122] Gazebo. *Control Description*. URL: %5Curl%7Bhttps://classic.gazebosim.org/tutorials?tut=plugins_hello_world&cat=write_plugin%7D.
- [123] Vries, Maarten de. *Switch to libb64 for base64 encoding/decoding*. URL: %5Curl%7Bhttps://github.com/ros/ros_comm/pull/1046%7D.
- [124] Görner, Michael and Ruppel, Philipp. *biotac_gazebo_plugin*. Version 1.0.0. URL: https://github.com/TAMS-Group/biotac_gazebo_plugin.
- [125] Melbye Staven, Victor. *biotac_sim_plugin*. Version 1.0.0. URL: https://github.com/vmstavens/biotac_sim_plugin.
- [126] E-MOTION. *ATI: 6-Axis Force and Torque Sensor (Nano17 Series) 9105-TW-NANO17-E-1.8*. URL: %5Curl%7Bhttps://www.e-motionsupply.com/product_p/9105-tw-nano17-e-1.8.htm%7D.
- [127] Olson, Edwin. “AprilTag: A robust and flexible visual fiducial system”. In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 3400–3407. doi: [10.1109/ICRA.2011.5979561](https://doi.org/10.1109/ICRA.2011.5979561).
- [128] Görner, Michael and Ruppel, Philipp. *Biotac Single Contact Response*. URL: %5Curl%7Bhttps://tams.informatik.uni-hamburg.de/research/datasets/index.php#biotac_single_contact_response%7D.
- [129] University, Stanford. *The Stanford 3D Scanning Repository*. <http://graphics.stanford.edu/data/3Dscanrep/>.
- [130] Parra, Álvaro and Chin, Tat-Jun. “Guaranteed Outlier Removal for Point Cloud Registration with Correspondences”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (Nov. 2017), pp. 1–1. doi: [10.1109/TPAMI.2017.2773482](https://doi.org/10.1109/TPAMI.2017.2773482).
- [131] Ranges. https://shadow-robot-company-dexterous-hand.readthedocs-hosted.com/en/latest/user_guide/md_ranges.html. Accessed: 2023-04-26.
- [132] Ngeo, Jimson, Tamei, Tomoya, and Shibata, Tomohiro. “Continuous and simultaneous estimation of finger kinematics using inputs from an EMG-to-muscle activation model”. In: *Journal of neuroengineering and rehabilitation* 11 (Aug. 2014), p. 122. doi: [10.1186/1743-0003-11-122](https://doi.org/10.1186/1743-0003-11-122).
- [133] Palmer, A K et al. “Functional wrist motion: a biomechanical study”. en. In: *J Hand Surg Am* 10.1 (Jan. 1985), pp. 39–46.

- [134] *Finger*. https://shadow-robot-company-dexterous-hand.readthedocs-hosted.com/en/latest/user_guide/md_finger.html. Accessed: 2023-04-26.
- [135] Mavrogiannis, Christoforos. “Grasp Synthesis Algorithms for Multifingered Robot Hands”. PhD thesis. Mar. 2013.
- [136] *Kinematics*. https://shadow-robot-company-dexterous-hand.readthedocs-hosted.com/en/latest/user_guide/md_kinematics.html. Accessed: 2023-04-26.

Appendix A

Shadow Dexterous Hand - Technical Specifications

Table A.1 shows the ROM for the Shadow Dexterous Hand and Table A.2 show the ROM for a human hand. The shorthand abbreviations used in these tables can be seen listed in Table A.3. The joints are numbered from fingertip to base and thus FF1 refers to the first joint after the fingertip on the first finger i.e. the index finger.

ROM - Shadow Dexterous Hand					
Joint(s)	Min deg	Max deg	Min rad	Max rad	Notes
FF1, MF1, RF1, LF1	0	90	0	1.571	Coupled
FF2, MF2, RF2, LF2	0	90	0	1.571	
FF3, MF3, RF3, LF3	-15	90	-0.262	1.571	
FF4, MF4, RF4, LF4	-20	20	-0.349	0.349	
LF5	0	45	0	0.785	
TH1	-15	90	-0.262	1.571	
TH2	-40	40	-0.698	0.698	
TH3	-12	12	-0.209	0.209	
TH4	0	70	0	1.222	
TH5	-60	60	-1.047	1.047	
WR1	-40	28	-0.698	0.489	
WR2	-28	10	-0.489	0.174	

Table A.1: The ranges of motion for each joint in the Shadow Dexterous Hand [131].

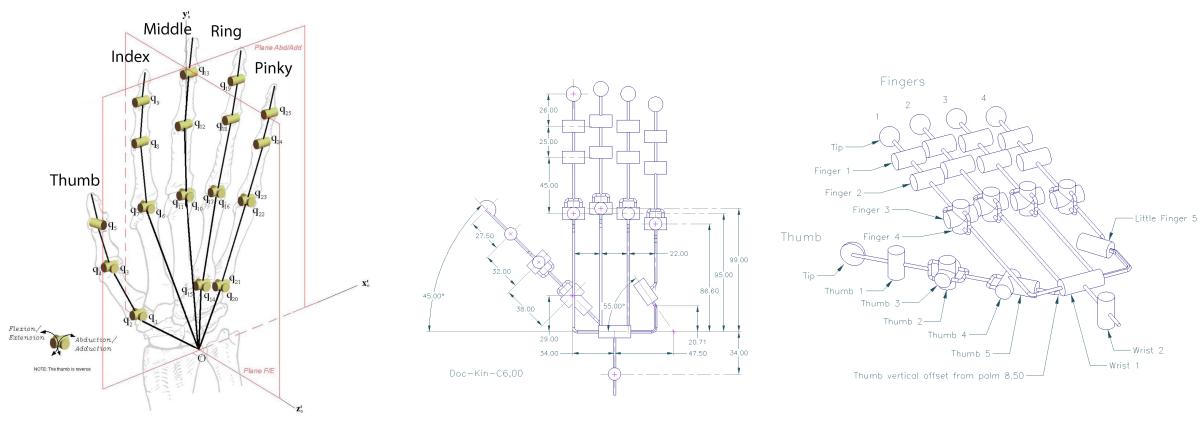
ROM - Human Hand					
Joint(s)	Min deg	Max deg	Min rad	Max rad	Latin Name
TH1	-15	80			Interphalangeal (IP)
TH2 + TH3	-10	55			Metacarpophalangeal (MCP)
TH4 + TH5	-10	55			Carpometacarpal (CMC)
FF1, MF1, RF1, LF1	0	80			Distal interphalangeal (DIP)
FF2, MF2, RF2, LF2	0	100			Proximal interphalangeal (PIP)
FF3, MF3, RF3, LF3	-45	90			Metacarpophalangeal (MCP)
WR1	-80	80			Radiocarpal
WR2	-28	20			Radiocarpal

Table A.2: The theoretical ROM for each finger joint in human hand [132] and found ROM for the wrist joints [133].

To compare the kinematic structure of the Shadow Dexterous Hand and a human hand, seen Fig. A.1.

Joint Name Abbreviation	
Abbreviation	Full Name
FF	First Finger
MF	Middle Finger
RF	Ring Finger
LF	Little Finger

Table A.3: The abbreviations used to reference [134].



(a) The kinematic tree of a human hand according to [135].

(b) The kinematic tree of the Shadow Dexterous Hand according to [136].

Fig. A.1: The kinematic trees of a human hand and the Shadow Dexterous hand.

Appendix B

Tactile Perception - Simulated Electrode Activations

Below three

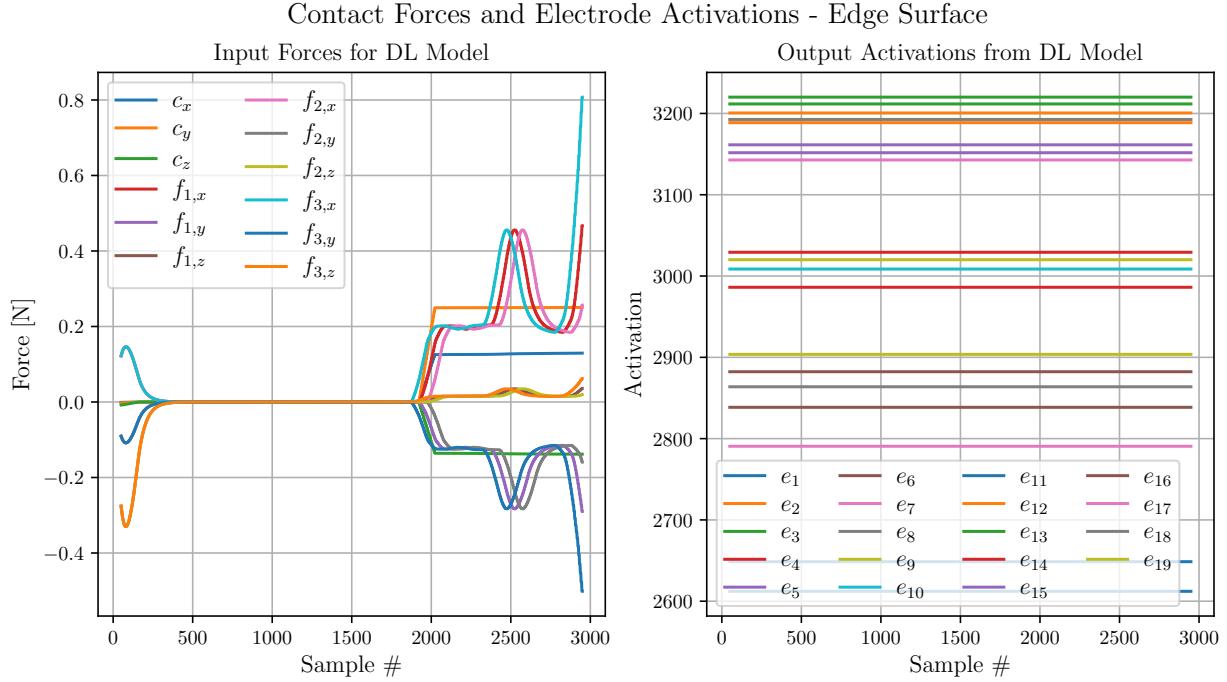


Fig. B.1: The simulated tactile electrode activations when the simulated Shadow Dexterous hand's index finger is in contact with an edge.

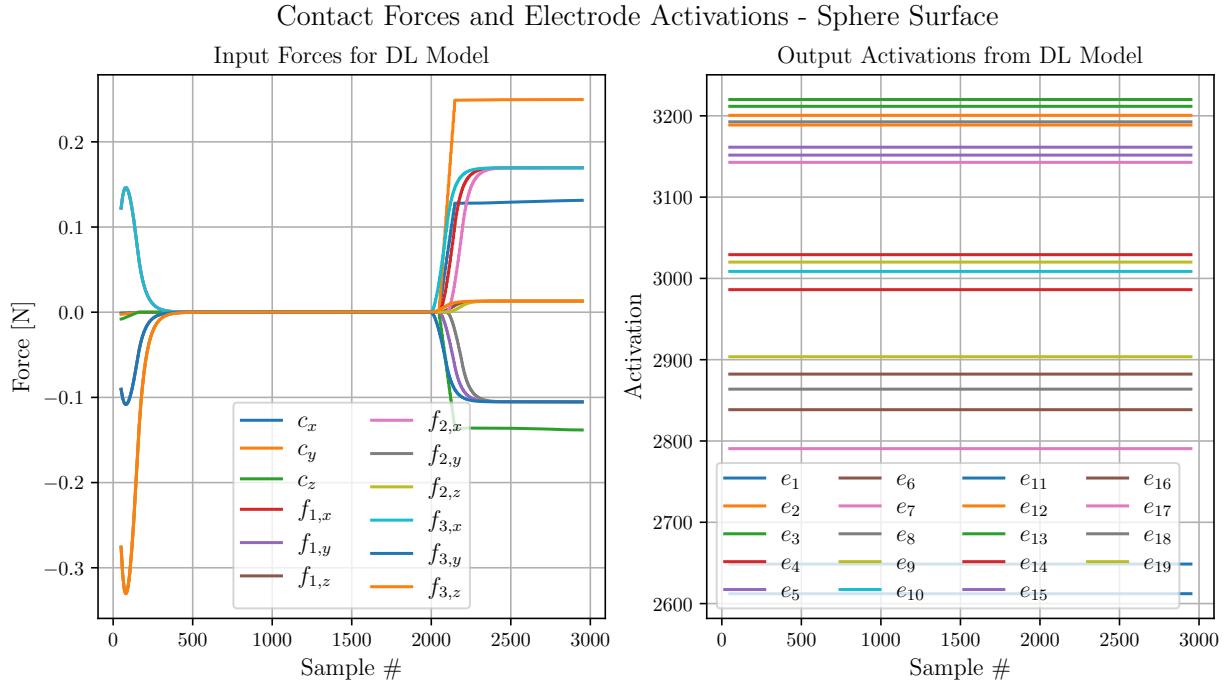


Fig. B.2: The simulated tactile electrode activations when the simulated Shadow Dexterous hand's index finger is in contact with a smooth surface.

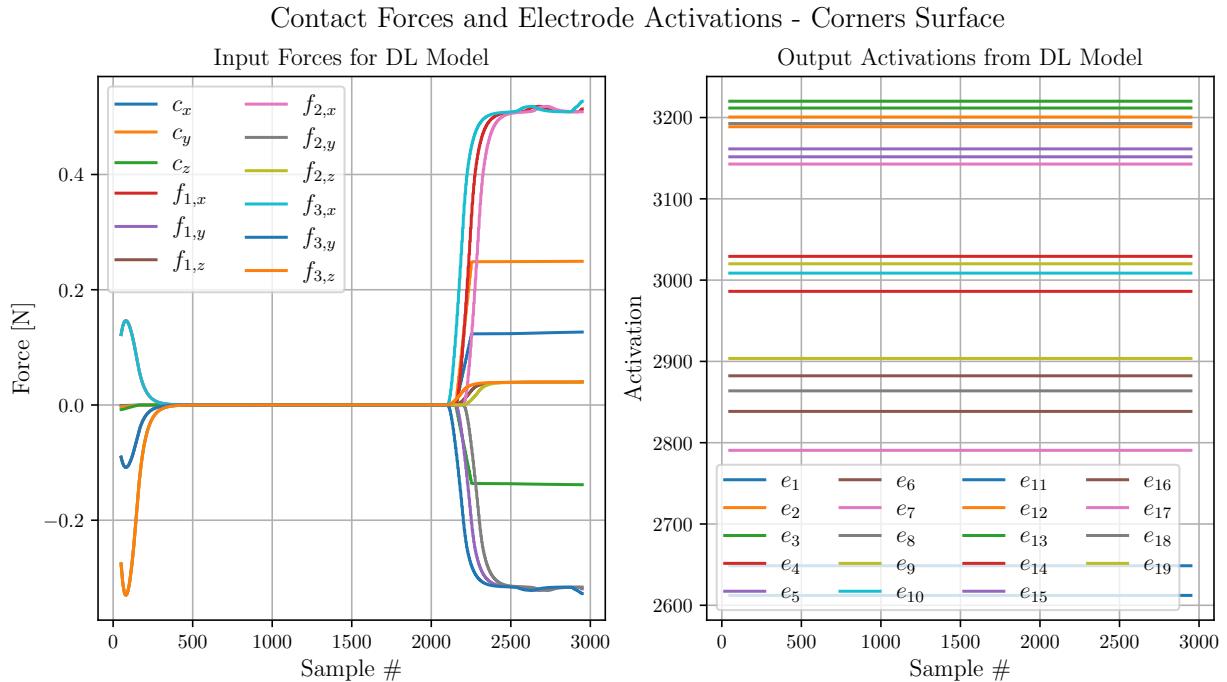


Fig. B.3: The simulated tactile electrode activations when the simulated Shadow Dexterous hand's index finger is in contact with a corner.