Mutual Capacitance Tactile Sensor Design and Sim-to-Real Bridging for Robotic Grasping

by

Abed Al Rahman AL MRAD

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT FOR A MASTER'S DEGREE
WITH THESIS IN AUTOMATED MANUFACTURING ENGINEERING
M.A.Sc.

MONTREAL, DECEMBER 21, 2023

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Jean-Philippe Roberge, Thesis supervisor
Systems engineering department, École de technologie supérieure

Mr. Georges Ghazi, Chair, Board of Examiners
Systems engineering department, École de technologie supérieure

Mr. Matthew Harker, External Examiner
Systems engineering department, École de technologie supérieure

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON "DEFENSE DATE: DECEMBER 13, 2023"

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

The completion of this master’s thesis signifies a profound milestone in my life, a testament to the relentless pursuit of knowledge and adventure. This journey has been a tapestry woven with transformative personal and academic experiences that have left an indelible mark on my character. I’ve been a student of life, learning from both successes and failures, encountering individuals who have illuminated my path, and those who’ve challenged me to grow. To all these experiences, I am eternally grateful.

I wish to commence my gratitude by acknowledging Jean-Philippe Roberge, whose unwavering support and guidance have been akin to that of an elder brother. Jean-Philippe, you’ve not only been an exceptional mentor but also a source of inspiration on both academic and personal fronts. It has been a privilege and an honor to be your student.

I extend my thanks to all the colleagues I have had the pleasure of interacting with during my studies.

To my cherished friends and beloved family members, your unwavering support and constant inspiration have been the bedrock of my journey. Your presence in my life is a source of immense gratitude.

As I reflect on this journey, I stand here with a sense of warmth and pride, knowing that this thesis represents not just academic achievement, but the embodiment of my growth and the promise of more adventures ahead.

Conception de capteurs tactiles à capacité mutuelle et réduction de l'écart simulation-réalité pour la préhension robotique

Abed Al Rahman AL MRAD

RÉSUMÉ

L'aptitude tactile des êtres humains est remarquable et elle est cruciale pour une multitude de tâches de la vie réelle. Transférer cette capacité exceptionnelle aux robots présente un grand potentiel dans diverses applications, en particulier dans des environnements dynamiques et inconnus où la vision par ordinateur n'est pas pratique. En conséquence, l'objectif de ce projet est de créer un capteur tactile économique capable de recouvrir entièrement un préhenseur robotique anthropomorphe et de tester sa fonctionnalité en évaluant sa capacité à représenter des formes primitives sous forme de nuages de points dans l'espace en 3D et à discerner ces formes. Ce projet représente une première étape vers la libération du potentiel futur permettant à un manipulateur robotique de fouiller dans des environnements encombrés, même en cas de capacités de vision par ordinateur limitées ou inexistantes. De plus, nous avons recréé la fonctionnalité du capteur dans un environnement de simulation pour reproduire des données tactiles synthétiques proches de celles générées dans la vie réelle pour une formation efficace des futurs modèles d'IA (*Intelligence artificielle*). Les modèles d'IA contemporains et à venir requièrent des larges ensembles de données pour un entraînement adéquat, et les simulations peuvent exploiter la puissance de la programmation parallèle, entre autres outils, pour accélérer et rationaliser le processus de génération de ces ensembles de données, améliorant ainsi son efficacité.

Après la conception du capteur tactile, nous l'avons monté sur un poste de travail robotique réel pour effectuer des expériences de préhension et de reconnaissance des formes. De plus, une réplique du système robotique a été utilisée en simulation pour générer des données tactiles synthétiques. Minimiser l'écart entre la simulation et la réalité est un objectif primordial garantissant que les données tactiles synthétiques générées reflètent étroitement les interactions du monde réel. Le capteur développé a réussi à capturer avec succès les formes de certains objets primitifs et à les discerner aussi bien en simulation qu'en réalité, et l'écart initial entre la simulation et la réalité a été considérablement réduit.

En résumé, nous avons développé un capteur tactile économique qui peut être intégré aux systèmes robotiques pour la reconnaissance des formes sans dépendre de la vision, et nous avons élaboré un cadre de simulation permettant de générer efficacement des données tactiles synthétiques réalistes qui pourront être utilisées pour former des algorithmes d'intelligence artificielle à l'avenir. Cependant, certaines modifications doivent être apportées pour améliorer la fiabilité et les performances du capteur tant en situation réelle qu'en simulation.

Mots-clés: capteur tactile, simulation, données tactiles synthetiques, reconnaissance de formes

Mutual Capacitance Tactile Sensor Design and Sim-to-Real Bridging for Robotic Grasping

Abed Al Rahman AL MRAD

ABSTRACT

The humans' tactile sensing ability is remarkable and it is crucial for a multitude of real-life tasks. Transferring this exceptional capability to robots presents a lot of potential across various applications, especially in dynamic and unknown environments where computer vision is not practical. As a result, the aim of this project is create a cost effective tactile sensor that can entirely cover an anthropomorphic robotic manipulator and test its functionality by assessing its ability to represent primitive shapes as point clouds in the 3D space and to discern these shapes. This project represents a first step towards unlocking the future potential of enabling a robotic manipulator to rummage in cluttered environments, even in instances of limited or absent computer vision capabilities. Furthermore, we recreated the sensor's functionality in a simulation environment to reproduce synthetic tactile data that are close to the ones generated in real-life for an efficient future AI models training. Contemporary and forthcoming AI models require large datasets for a proper training and simulations can leverage the power parallel programming among many other tools to expedite and streamline the datasets generation process , thereby enhancing its efficiency.

After designing the tactile sensor, we mounted it on a real-life robotic workstation to perform objects grasping and shape recognition experiments.Furthermore, a simulation replica of the robotic system was to generate synthetic tactile data. Minimizing the sim-to-real gap is a paramount objective, ensuring the generated synthetic tactile data closely mirrors real-world interactions.

The developed sensor was successfully able to capture the shapes of some primitives objects and to discern them in both simulation and reality and the initial sim-to-real gap was significantly reduced.

In sum, we have developed a cost-effective tactile sensor that can be integrated with robotic systems for shape recognition without relying on vision and we have developed a simulation framework to efficiently generate realistic synthetic tactile data that can be used to train AI (*Artificial intelligence*) algorithms in the future. However, some modifications should be implemented to improve the reliability and performance of the sensor in both real-life and simulation.

Keywords: tactile sensor, simulation, synthetic tactile data, shape recognition

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AI	Artificial Intelligence
AlN	Aluminum Nitride
AoA	Angle of Approach
BDML	Biomimetics and Dextrous Manipulation Lab
CAD	Computer-aided Design
CAN	Controller Area Network
CNN	Convolutional Neural Network
CoRo	Control and Robotics
CPU	Central Processing Unit
C_M	Mutual Capacitance
CSV	Comma Separate Value
CSX	Capsense Crosspoint
DoF	Degrees of Freedom
FEA	Finite Element Analysis
FEM	Finite Element Method
FBG	Fiber Bragg Grating
FPCB	Flexible Printed Circuit Board

IC	Integrated Circuit
ICP	Iterative Closest Point
I2C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
LED	Light Emitting Diode
MDP	Markov Decision Process
MEMS	Micro-Electro-Mechanical Systems
MIS	Minimally Invasive Surgeries
MRI	Magnetic Resonance Imaging
N	Number of points to use for mean distance estimation
nSigma	Standard Deviation Multiplier
ODE	Open Dynamics Engine
PDMS	Polydimethylsiloxane
PID	Proportional-Integral-Derivative
PSF	Point Spread Function
PSoC	Programmable system-on-chip
PVDF	Polyvinylidene Fluoride
RL	Real-life
ROS	Robot Operating System
RX	Receiver

SCL	Serial Clock
SDA	Serial Data
SDF	Simulation Description Format
SIM	Simulation
SNR	Signal to Noise Ratio
SOR	Statistical Outlier Removal
Std	Standard Deviation
SVM	Support Vector Machine
TIR	Total Iternal Reflection
TX	Transmitter
UART	Universal Asynchronous Receiver-Transmitter
UR	Universal Robots
URDF	Unified Robot Description Format
USB	Universal Serial Bus

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

cm^2	Centimeter squared
°C	Degrees Celsius
Hz	Hertz
mm	Millimeter
N	Newton
USD	United States Dollar

INTRODUCTION

Humans possess an astonishing ability to employ tactile sensing for a multitude of tasks including object manipulation and recognition, even in cluttered environments. An illustrative example of this proficiency is when individuals effortlessly identify items concealed within their pockets through touch alone. This innate capability showcases the finesse of human tactile perception that enables us to navigate and interact with our surroundings effectively.

Transferring this exceptional tactile sensing capability to robotic systems holds immense promise across various applications. In the realm of manufacturing, robots equipped with tactile sensors can delicately handle objects with dexterity, ensuring precision and quality control. Moreover in collaborative work environments where humans and robots cooperate, tactile sensors could play a pivotal role in ensuring safety. These sensors enable robots to sense human presence and adjust their actions accordingly, mitigating potential risks. Furthermore, despite its significant strides in robotics, the limitations of computer vision become apparent in dynamic and unknown environments, especially those that are partially occluded, obscured, or lack adequate lighting. In such scenarios, relying solely on visual perception is impractical and unreliable. This is where tactile sensing emerges as a complementary and, in some cases, indispensable sensory modality that equips robots with the adaptability to navigate and manipulate objects effectively, or in other words it provides robotic systems with the ability to "feel" their surroundings. Therefore integrating tactile sensing capabilities into robotic systems partly bridges the perceptual gap and enhances their autonomy and versatility.

As a result, the aim of this project is to create a cost effective tactile sensor that can entirely cover a relatively large, anthropomorphic robotic gripper with 16 DoF (*Degrees of freedom*) and test it for shapes identification. This project is a first step towards allowing a robotic manipulator to rummage in cluttered environments and reach a target object with limited or no visual perception. As a first step, a design for a capacitive tactile sensor that leverages the fringe effect in capacitors is proposed. Unlike other tactile sensors in the market nowadays like optical-based ones such as Gelsight and magnetic-based ones, our particular design is cost effective since it requires easily-available materials and a relatively simple production process as will be shown in details in chapter 4. Moreover, the introduced sensor can be easily applied to an entire relatively large robotic gripper and this offers a better perception which is particularly useful for rummaging in cluttered environments.

Another important aspect that was considered in our project is simulations, which stand as a cornerstone in the realm of robotics, offering an indispensable resource for researchers and engineers engaged in the development and advancement of robotic systems, particularly in the area of synthetic data generation and this significance emanates from multifaceted advantages that simulations bring to the forefront. Firstly, simulations grant an unprecedented degree of flexibility, enabling the comprehensive exploration of diverse scenarios and environmental conditions, spanning a broad spectrum of parameters and contexts, without the constraints associated with physical prototypes. For instance, domain randomization is a well known approach in the field of robotics which involves the deliberate introduction of variability and uncertainty into simulated environments and this is used for training robotic algorithms and machine learning models that require large training datasets. Factors such as lighting conditions as well as object appearances, friction coefficients and physical properties can be randomized within these virtual worlds and therefore allowing the trained robotic algorithms to better generalize their learning from simulated environments to the unpredictability of the physical world. Secondly the concept of parallel programming can be applied to train multiple robots simultaneously in different simulation environments and therefore speeding up the overall needed training time. This heightened versatility results in both time and cost savings, essential considerations in research and development endeavors.

Ergo, after mounting the developed sensor on a robotic manipulator and driven by previously discussed advantages of simulation in robotics, we decided to create a simulation replica of our robotic system with the aim to accelerate the tactile data collection for future training of AI (*Artificial intelligence*) algorithms while providing a cheaper and more convenient data collection approach. Therefore, as another objective followed up for this project, we aimed for minimizing the sim-to-real gap as much as possible to make our simulations generate realistic synthetic tactile data which can further boost the performance of the future AI algorithms.

In this thesis we first perform a literature review in chapter 1 to introduce the different transduction mechanisms used for tactile sensors by the researchers in the robotics field nowadays and we compare them to our selected mechanism and design. We also review the state of the art approaches for tactile sensors simulation and synthetic data generation. Then, in chapter 2 we go over the theory and operating principles of the proposed tactile sensor. In chapter 3, we describe in details our sensor design and we go over the hardware setup. Afterward, we discuss our simulator selection and simulation environment setup in chapter 4. Finally, we describe our experimental setup and our analysis for the generated results in chapter 5 before concluding and opening on potential future research avenues.

CHAPTER 1

LITERATURE REVIEW

Before discussing our developed sensor, we present the state of the art research work that has been recently done in the tactile sensing field. In this chapter, we first go over the different transduction mechanisms that were developed for tactile sensing while assessing each work and comparing it to our project. Furthermore, since the generation of synthetic tactile data through simulation was a major focus of our project, we cover the recent research in the area of tactile sensors simulation that spans different simulators and simulation techniques or approaches.

1.1 Tactile Transduction Mechanisms

Over the past couple of decades, a variety of tactile sensors based on different transduction mechanisms have been developed. The major technologies in this regard can be mainly classified into capacitive, piezoresistive, magnetic, optical, vision-based, piezoelectric and barometric sensors.

1.1.1 Capacitive Tactile Sensors

This type of sensors relies on the variation of capacitance to measure the applied pressure/force. For instance, in the case of parallel plate capacitors, the capacitance can be expressed as in equation 1.1.

$$C = \epsilon_0 \epsilon_r \frac{A}{d} \tag{1.1}$$

where ϵ_0 is the vacuum permittivity, ϵ_r is the dielectric layer relative permittivity, A is the overlapping area between the two electrodes and d is the distance between the electrodes. Therefore, a force applied to the sensor leads to a change in A and/or d and various designs of the dielectric layer and the electrode structure lead to different types of capacitive sensors with unique capabilities. For instance, a compressible dielectric layer is commonly used to promote the sensitivity of the capacitive sensors. Mannsfeld *et al.* (2010) have developed a capacitive tactile sensor with a compressible dielectric layer to create unprecedented short response time and high sensitivity. The permittivity variation that results from the deformation of the pyramids in the developed PDMS (*Polydimethylsiloxane*) based pyramid-microstructured dielectric allowed to detect ultra-small pressures. Despite their high performance, these sensors cannot measure shear forces. To realize the measurement of the normal and shear components of external three-axis forces, Liang, Wang, Mei, Xi & Chen (2015) developed a four capacitors based sensor unit that translates the normal and shear forces into capacitance difference. In this study, the previously discussed pyramid-microstuctured PDMS from Mannsfeld *et al.* (2010) was used as a dielectric but the pyramid tips were truncated to improve the capacitors robustness to shear forces while maintaining a relatively high sensitivity and short response time. There are other designs for the dielectric layer that have been tested such as nano-needles (Kim, Nga Ng & Soo Kim, 2012) and fluid (Ridzuan, Masuda & Miki, 2012). While these designs offer highly sensitive sensors, the response time is very high since the used materials take a relatively long time to recover to their initial states. Moreover, several electrodes designs were proposed to increase both robustness and sensitivity. For instance, Dobrzynska & Gijs (2012) developed a polymer-based capacitive sensor with finger-shaped electrodes to increase the sensitivity to shear forces. Surapaneni, Xie, Guo, Young & Mastrangelo (2012) created a three-axis capacitive sensor with electrodes that float in a PDMS layer on top of two sets of electrode lines integrated inside a FPCB (*Flexible printed circuit board*): drive and sense lines. While the four capacitors created by this setup experience the same capacitance variation when pure normal forces are applied however the shear forces create a shift in the overlap with the floating electrodes. It was claimed that this design improved the capacitive sensing reliability, robustness and shear sensitivity. Another work that is worth mentioning was done by Sicotte-Brisson, Bernier, Kwiatkowski & Duchaine (2022). The research paper focuses on the development and characterization of a novel capacitive tactile sensor that relies on the mutual capacitance technology to detect normal forces in a 2D (*Two dimensional*) array for robotics applications. The electrodes layout in the proposed design follows a grid like structure and a shielding fabric was used to provide a sensor's isolation and prevent undesirable electrical interference from external conductive objects. Based on their conducted experiments, according to the authors the proposed design offers a superior spatial resolution without compromising the sensor's sensitivity, as well as operational range and frequency. However, unlike our proposed design the sensor was printed on a rigid PCB the thing that limits its flexibility and compliance with the robotic surfaces. Also, the authors acknowledge the need to further optimize the sensor's dielectric and shielding fabric flexibility. Finally, Ruth *et al.* (2021) introduced a fringe effect sensor with an interdigitated electrode design mounted by a microstructured dielectric. According to the authors, the developed sensor combines both pressure and proximity sensing without compromising the performance in one for the other. In more details, a thorough analytical and numerical analysis of various capacitive sensors with different designs was performed in this paper, including a parallel plate sensor and a fringe field sensor with various electrodes layouts. The fringe field sensors, which capitalize on the fringe effect caused by disruptions in the electric field, demonstrate a remarkable sensitivity to approaching objects in the noncontact mode. However, the authors proposed an interdigitated electrodes layout to enhance the noncontact sensitivity by increasing the electric field strength of the tactile sensor. As a result, the sensor was able to distinguish between conductive and nonconductive objects in noncontact mode. Furthermore, the paper achieves a breakthrough by demonstrating that electrode interdigitation significantly enhances pressure sensing in contact mode making it suitable for human touch-like applications. Finally, in their work Ruth *et al.* (2021) integrated the interdigitated sensor into a robotic gripper and showcased its exceptional proximity and pressure sensing capabilities. This sensor design was adopted in our project as part of our collaboration with the BDML (*Biomimetics and dextrous manipulation lab*) at Stanford university. Furthermore, it is worth mentioning that our developed sensor was employed by our collaborators from Stanford for object motion classification and this work was presented in a research paper by Thomasson, Roberge, Cutkosky & Roberge (2022). In this work, the sensor was installed only on the back of an allegro hand and was used to formulate tactile cues and infer the object motion during incidental contacts. According to the authors, this allows a robotic manipulator to safely navigate in cluttered environments and rearrange objects with no explicit motion planning or object recognition. While with the proposed approach many classification challenges are met however the method has shown high accuracy levels for prototypical objects with simple geometries.

1.1.2 Piezoresistive Tactile Sensors

These sensors transform external forces into resistance changes. The most common approaches in this area include nanocomposites, strain gauges and doped silicon cantilever beams. Nanocomposites-based sensors rely on the advancement of nanomaterial and microstructure fabrication and are generally composed of soft polymer matrices with embedded nanoscale conductive fillers that can be either metal-based or carbon-based. Such sensors can be classified into two categories: those that rely on materials volumetric changes (Castellanos-Ramos, Navas-González, Fernández & Vidal-Verdú, 2015) and those that use the change in the contact area at the microscopic scale (Strohmayr, Wörn & Hirzinger, 2013) Strain gauge on the other hand is composed of a patterned metallic foil printed on a flexible substrate. Such a sensor will stretch or compress with the structure over which it is placed therefore capturing mechanical deformations and transducing them into electrical resistance change. Strain gauges are typical for contact force measurements (da Silva, de Carvalho & da Silva, 2002). A doped silicon beam is another type of piezoresistive sensors that is more sensitive and provides a wider measurement range compared to strain gauges but both sensors suffer from the fragility of silicon material and the inability to conform to curved surfaces. Takahashi, Nakai, Thanh-Vinh, Matsumoto & Shimoyama (2013) developped a triaxial tactile sensor using doped silicon beams for both normal and shear force sensing.

In a recent work, Wang, Lu, Mei & Zhu (2020) presented a wearable tactile sensor that relies on galinstan liquid metal for a simultaneous force and temperature measurement. The authors introduced a unique design that includes a fingerprint patterned microfluidic channels and a top oval-shaped protrusion to increase the sensor's force sensitivity. Moreover, a Wheatstone bridge circuit is employed for force and temperature signals decoupling. While it is reported that the presented sensor is cost-effective and demonstrates a good repeatability and a high force sensitivity down to 0.32 N (*Newton*), a limited measurement range for both force and temperature was presented in this paper (i.e. 0-13.5 N and up to 80°C (*Degrees Celsius*)and it is worth mentioning that the sensor's performance can be influenced by different factors such as electromagnetic interference, humidity and mechanical wear and tear. For our developed sensor the force range as will be later presented in chapter 3 is much higher and can reach 50 N and light forces down to 0.06 N can be detected. Moreover, Zhao *et al.* (2021) proposed a flexible dual-parameter tactile sensor that combines a pressure-sensitive module and a strain-sensitive module to simultaneously measure both the magnitude and direction of the applied force. On one hand, the pressure-sensitive module is made from 3D (*Three dimensional*)tubular graphene sponge, which is a highly compressible material whose electrical resistance varies significantly upon the application of external forces. On the other hand, the strain-sensitive module is made from a hybrid nanocomposite of silver nanowires and carbon nanotubes and is built with spider's web like structure that can stretch in specific directions when external forces are applied leading to electrical resistance variations that can detect the forces directions. The authors have successfully achieved a resistance matching between the two combined modules and therefore enabling highly accurate force measurements and the sensor's high performance and force monitoring capabilities were clearly demonstrated. Nonetheless, the presented work needs additional sensor characterization and performance analysis under different environment conditions. Also, the fabrication processes proposed in this paper like chemical vapor deposition and spray embedding are complicated and require special equipment and expertise. However the material used in our project are readily available and the proposed fabrication methods are mainstream and scalable.

While piezoresistive sensors present excellent performance results, hysteresis and lack of reproducibility remain the major challenges. Moreover, in some cases these sensors are expensive to fabricate compared to our proposed interdigitated mutual capacitance based tactile sensor.

1.1.3 Magnetic Tactile Sensors

This is another type of sensors that relies on magnetism to detect contact forces. Known for their robustness and their lack of mechanical hysteresis, these sensors can detect external forces that lead to mechanical deformation mainly through one of two operating principles: Either the measurement of magnetic field intensity variation by the Hall effect or the use of the electromagnetic induction principle. For instance, Ledermann, Wirges, Oertel, Mende & Woern (2013) developed a magnetic tactile sensor using a permanent magnet and a 3D (*Three dimensional*) Hall sensor AS54xx. The permanent magnet is embedded in an elastic material such that when external forces are applied, the elastic material will deform and the permanent magnet position will change which leads to a distortion in the magnetic field vector that can be captured by the AS54xx underneath. Given the big size of the developed sensor prototype, which negatively affects the spacial resolution of tactile sensing, Alfadhel, Khan, de Freitas & Kosel (2016) developed a more compact tactile sensor that can capture normal and shear force and therefore allowing the recognition of small changes in surface texture. The sensor is composed of nanocomposite cilia placed on magnetic micro-sensors. The nanocomposite is a mix of PDMS and iron nanowires that result in permanent magnet properties. It was shown that the sensor has extremely low power dissipation and can measure 3D forces in liquid environments. On the other hand, Wattanasarn, Noda, Matsumoto & Shimoyama (2012) embedded flexible induction coils in elastomeric substrates to create a 3D tactile sensor. The coils are embedded in two layers that sandwich a PDMS spacer layer. The coils underneath the PDMS are called excitation coils and an alternating current passes through them to generate a magnetic field. This field is detected by the coils near the external surface of the sensor, which are called detection coils and as a result they generate an induced voltage. The voltage variation in the detection coils whenever a mechanical deformation of the sensor is caused by external forces is the key behind this sensor's functionality. Moreover, Bhirangi, Hellebrekers, Majidi & Gupta (2021) proposed ReSkin, a novel magnetic-based soft tactile sensor for robotic applications. The authors present a design that combines machine learning with magnetic sensing technology to provide a long lasting, versatile and high resolution tactile sensor. Reskin can be molded into different shapes and thicknesses since it consists of a soft magnetized skin that is embedded with magnetic micro-particles and and a flexible magnetometer-based sensing mechanism. Under external forces, the relative distance between the magnetic particles and the magnetometer changes leading to a detectable variation of the magnetic flux. The authors demonstrated the robustness of developed sensor to wear and tear, its adaptability through the integrated machine learning models and its effectiveness to diverse real-life applications. Magnetic tactile sensors can achieve high spacial resolution and they present a low power consumption alternative compared to other sensors, especially when permanent magnets are used. However, there performance can be hampered by parasitic effect like eddy current and stray capacitance in the case of electromagnetic induction sensors.

1.1.4 Optical Tactile Sensors

Optical sensors relate tactile data to an output light changes. These sensors that typically employ optical fibers as the medium for light transmission, operate most commonly using one of two principles: light intensity modulation and FBG (*Fiber bragg grating*). As the name implies tactile sensors based on light intensity analysis infer tactile data by measuring the variation in the intensity of the light coming out of the fiber optics, which occurs because of fibers bending under external contact forces. Ahmadi, Packirisamy, Dargahi & Cecere (2011) developed an optical tactile sensor that detects the position of the contact forces by measuring the power loss in each of the fibers but the power used by the sensor as well as its size were inadequately high. Xie, Jiang, Seneviratne & Althoefer (2012) proposed a mirror based optical sensor array that measure normal contact forces. When a normal force is applied on a sensing element the correspondent fiber's light intensity increases and the related pixel in the output camera video gets activated. On the other hand, the FBG technology consists of placing gratings in the optical fibers, which diffracts the transmitted light and filters out specific wavelengths at a time depending on the effective refractive index of the optical fiber core and the spatial period of the grating. External forces applied to the sensor affect the latter factors and therefore cause a shift in the filtered wavelengths. Ledermann, Hergenhan, Weede & Woern (2012) developed FBG based tactile sensors for MIS (*Minimally invasive surgeries*). The developed sensor analyzes the reflected light spectrum to detect minimal external strains. Moreover, in their recent work Li *et al.* (2022) developed an optical-based tactile sensor that can detect normal forces on 3D curved surfaces that are typically encountered in surgical instruments during MIS. The proposed sensor operates based on the optical waveguides and TIR (*total internal reflection*) principles. More specifically, the sensor is composed of an optical fiber core and a surrounding cladding. During normal state, the light transmitted through the core remains confined within this core, but when an external force is applied, the subsequent core's deformation leads to the TIR phenomenon that dictates a light leakage from the sensor's core to the surrounding cladding. Since there is a direct relationship between the amount of escaping light and the amount of undergone deformation, the change in the light's intensity at the sensor's output can be used to measure the magnitude of the applied force. The authors have validated the the proposed sensor's design through simulation and the sensor displayed a high precision, a high resolution of 0.026 N as well as an excellent repeatability, a low hysteresis and a rapid response time, making it effective for scenarios involving MIS. However the sensor's detection is limited to normal forces and does not address shear forces which restricts its range of applications. Also, the sensor's fabrication involves some complicated procedures which can increase the manufacturing costs and can pose some scalability challenges.

Optical tactile sensors have a small response time and can be small in size in addition to being immune to electromagnetic interference, which makes them compatible with MRI (*Magnetic resonance imaging*) and therefore suitable for an array of medical applications. However researchers claim that these sensors are susceptible to temperature influence, can suffer from light loss due to fibers micro-bending and the light analysis to extract tactile information is relatively speaking computationally expensive.

1.1.5 Vision-based Tactile Sensors

In this review, vision-based tactile sensors are distinguished from optical-based ones. While optical-based sensors rely on the analysis of light properties to detect contact and infer its properties, vision-based sensors involve the use of a camera to depict the area of contact and analyze it to infer the contact information. It is worth noting that vision-based tactile sensing is a hot research area nowadays because they provide an improved robotic perception compared to other tactile sensor and they present a great potential. One of the earliest and well-known vision-based tactile sensors is GelSight that was developed by Yuan, Dong & Adelson (2017). The core of the proposed sensor is made from a compliant elastomer layer coated with a reflective material and illuminated by LEDs (*Light emitting diodes*) that are placed underneath. The elastomer layer conforms to the external object's shape upon contact and a reflection that captures the object's surface details is created. A camera is placed beneath the elastomer to capture images of this reflection which are analyzed by image processing algorithms to measure the applied forces and torques as well as reconstruct the object's 3D shape and to identify other tactile properties. The authors claim that the proposed sensor provides high-resolution tactile data. It can also measure both normal and shear contact forces and can recognize different materials based on the perceived surface texture. However, the fabrication process of GelSight is complicated which makes its mass production challenging and expensive. Also, the sensor has a small field of view limited to 60 mm (*Millimeter*) x 40 mm which restricts its application just small object and tasks that requires a narrow view of the environment. Another work by Donlon *et al.* (2018) and that was inspired by GelSight sensing techniques, presented an enhanced vision-based tactile sensor known as GelSlim. Similarly to GelSight, the proposed sensor capture contact data by analyzing the images of a camera placed under an illuminated elastomer layer. However, the main focus of the authors was to introduce a more compact and a more durable tactile sensor than the original GelSight. To acheive this goal Donlon *et al.* (2018) integrated acrylic wave guides and mirrors into their design for a more efficient direction and control of the LEDs light while maintaining the quality of the reflection image. Moreover a parabolic reflection technique was employed to direct the light rays onto the elastomer in a parallel way which minimizes light intensity loss and ensures a consistent illumination. This optical path optimization allows for a better placement of the camera and the LEDs to allow for a slimmer sensor and a more compact design. Furthermore, according to the authors the elastomer layer in GelSight is covered with a textured fabric skin which provides more protection and durability while enhancing the sensor's sensitivity. However, despite these improvements the fabrication procedure of GelSlim is still complicated and requires specific tools and skill sets which leads to a higher manufacturing cost. GelSight wedge is another vision-based tactile sensor that was inspired by GelSight and it was proposed by Wang, She, Romero & Adelson (2021a) with a higher compactness and enhanced perception resolution. The proposed design has even a higher compact form factor than the previously discussed GelSlim and the authors claim that they down scaled the sensor to match the size of a human finger. Moreover, the authors addressed the problem of missing lights in some configurations by employing a trained neural network for gradient estimation, the thing that along with the versatility in the lighting configurations that GelSight wedge has to offer leads to an improved image gradients calculation and 3D shape reconstruction. However just like its predecessor sensors, GelSight wedge is still complicated to fabricate and some factors like shadows can affect the accuracy of the 3D shape reconstruction, not to mention the high computational cost for real-time operation especially after integrating a neural network for image gradient estimation. Another interesting work by Padmanabha *et al.* (2020) introduced OmniTact, a novel vision-based tactile sensor that provides a high-resolution and multi-directional tactile sensing on curved surfaces. While the proposed sensor shares the same fundamental principles of the previously discussed GelSight sensor and its derivates, OmniTact has a curved outer surface that is equipped with multiple strategically positioned micro-cameras allowing it to depict the elastomer deformations from different angles and directions. This added feature makes a major difference since it allows for an interaction between robots and external objects from different angles and positions. In this paper, the authors have used CNNs (*Convolutional neural networks*) to demonstrate the sensor's effectiveness in tactile-based control tasks. For instance, OmniTact was able to successfuly able to insert an electrical connector into an outlet by solely relying on tactile feedback. Despite OmniTact's improved sensitivity and its multi-directional contact detection ability, the authors acknowledge the high cost of the cameras and the complexity of the fabrication process as a limitation and propose alternative approaches to increase the sensor's affordability. As a further step towards improving the performance of vision-based tactile sensors while reducing their manufacturing complexity, researchers from Facebook have introduced DIGIT as a low cost and high resolution tactile sensor for in hand robotic manipulation. While the sensor's operational principles are the same as the previously discussed vision-based tactile sensors, in their paper Lambeta *et al.* (2020) present many added advantages of their proposed design. For instance, DIGIT is composed of a three-pieces plastic enclosure that encapsulates the camera, the LEDs, the elastomer and the rest of the sensor's components. This modular design allows for an ease of assembly, the thing that simplifies maintenance and leads to a more streamlined fabrication process which in its turn results in a lower sensors' cost. At the time of writing the paper, the authors claim that DIGIT costs around 15 USD (*United States dollar*) per sensor when produced in batches of 1000. This is half the price of the Gelsight sensor (Lambeta *et al.*, 2020). Moreover, the authors have integrated a video predictive model with the sensor for tactile data processing and a model predictive control for in-hand objects manipulation based on tactile feedback. These features allow DIGIT to anticipate the future elastomer deformations based on the current ones and to learn and adapt to the rolling dynamics of object manipulation. Finally, it is worth mentioning that CoRo (*Control and robotics*) lab has been also involved with vision-based tactile sensors recently, and StereoTac is one of these sensors that was presented by Roberge, Fornes & Roberge (2023). The proposed sensor presents a novel combination of tactile sensing and 3D vision to improve robotic manipulation and perception. Similarly to the previously discussed GelSight and its derivative sensors, StereoTac relies on the photometric stereo technique to reconstruct the shapes of the objects that enter in contact with its semi-transparent membrane. However, the proposed sensor has stereoscopic vision capabilities that rely on the incorporated two 2D neighbouring cameras which allows for capturing a 3D representation of the external environment before contact. Therefore, StereoTac overcomes this traditional vision-based tactile sensors' limitation. Despite its added value, StereoTac presents many limitations that should be addressed. For instance, the 14 mm spacing between the 2D cameras dictates a limited depth range which might affect the depth perception accuracy, especially for objects lying at longer distances.Moreover, the authors acknowledge the influence of noise on the quality of the generated data and this needs to be addressed. Finally, the current design is bulky and needs to be significantly more compact to be suitable for a variety of robotics applications like rummaging in cluttered environments.

For this project, the proposed capacitive tactile sensing design allows to cover the whole surface of an anthropomorphic robotic gripper using FPCB based tactile sensors for an increased perception, which cannot be practically achieved using the previously discussed optical and vision-based sensors because of their relatively complex setup. Moreover, our proposed design allows for simultaneous contact and proximity or non-contact sensing which allows for the identification of a wider range of surrounding objects properties including material and shape among others.

1.1.6 Other Transduction Mechanisms

Other adopted transduction mechanisms for tactile sensing that were found in the literature include the piezoelectric-based and the barometric-based ones. For instance, in their paper Yeo, Jung, Sim, Jang & Choi (2020) discuss the development of an AlN (*Aluminum Nitride*) based tactile sensor array using MEMS (*Micro-electro-mechanical systems*) technology. The proposed sensor array consists of a compact area of 2.5 x 2.5 cm^2 (*centimeter squared*) over which 2304 tactile sensor units were integrated with an AlN thin film as the core of each individual sensor and PDMS supporting layers . AlN is a piezoelectric material that converts the external forces applied to its surface into a proportional electrical signal that can be processed to identify the magnitude and distribution of these forces. The authors claim that their sensor exhibits a high flexibility, a high sensitivity allowing it to detect forces as low as 0.2 N and a negligible crosstalk among the sensor's electrical channels. However, the range of force applied to the sensor in this paper was limited with values between 0.2 N and 1.2 N which is not sufficient for many robotics applications and the authors mentioned that the sensor's output signal falls within the electrical noise range sometimes which affects the accuracy of the sensor's response. Moreover, the used materials in the presented design like AIN are expensive and the used micro-fabrication processes such as photolithography and reactive sputtering are complex and poses significant challenges especially in mass production. Furthermore, Li, Yin, Wee, Chinnappan & Ramakrishna (2023) have focused on the development of a piezoelectric tactile sensor using a nanofibrous membrane made from the PVDF (*Polyvinylidene fluoride*) material. The authors have also successfully improved the sensor's performance by adding carbon nanotubes and barium titanate to the PVDF membrane. The resultant sensor displayed a high sensitivity over a wide range of pressures and it also exhibited a great durability and a rapid response time. However, the authors have tested the developed tactile sensor only for human motion monitoring and recognition. While this is a remarkable application, this sensor can be used and tested in other fields such as robotics.

Furthermore, in this paper the researchers have tested the sensor only in a controlled laboratory environment and a real-world validation is necessary to confirm the sensor's utility.

In general, piezoelectric-based tactile sensors are expensive and require complex fabrication procedure, but in our project, we employed readily accessible materials and commonly used fabrication methods.

Another type of tactile sensors that deserves to be mentioned is the barometric-based ones. In their work, Kõiva *et al.* (2020) presented a novel tactile sensors array that relies on barometers to detect contact on the Shadow dexterous hand. Barometers are normally used to measure atmospheric pressure but in this work, upon the application of external forces on the robotic hand the barometers' membranes deform which changes the pressure within the sensors and these pressure variation are converted into electrical signals. Therefore, by analyzing the pressure distribution across the developed tactile sensor array, the distribution and magnitude of the forces applied on the shadow hand can be identified allowing the robotic hand to interact with its surrounding. The authors have used a readily available barometers from Bosch sensortec and they modified their orifices to enhance their performance. The modified sensors array exhibited a high stability and robustness and was able to classify the stiffness of different objects using a recurrent neural network with an 80% accuracy. However, the presented sensor array does not cover the whole robotic hand and the electronic setup is not well optimized and it was mentioned in the paper that the sensors array integration on the robotic hand can be further improved by eliminating external connections and adding more arrays to cover the remaining parts of the shadow hand. While the barometric-based tactile sensors can be accurate and reliable over a wide range of external pressures in a contact mode, unlike our developed sensor they do not have the potential to distinguish different material properties in a non-contact mode such as electrical conductivity. Moreover for this project, capacitive tactile sensing was used because as previously discussed it allows to cover the whole surface of a robotic hand, which cannot be practically achieved using other more complicated transduction mechanisms. Also, a relatively high spatial resolution can be achieved using the capacitive technology which allows to draw more semantic data from the robotic hand interaction with its surrounding, the thing that is very convenient for rummaging cluttered environments which is the long-term ultimate goal of this project in the future.

1.2 Tactile Sensors Simulation and Synthetic Data Generation

In order to equip the robotic manipulators with advanced grasping and environment navigation capabilities, many AI algorithms in the realm of reinforcement, supervised and unsupervised learning are essential. However, the reliability and robustness of an AI algorithm is heavily influenced by the amount of data available for training. In most cases, the complexity and the high expenses of the hardware setups needed to collect the needed data, push the researchers to redirect their efforts toward synthetic data generation, which consists of collecting all or part of the training data virtually in simulation environments. Given the highly accessible computational resources nowadays and the ability to run multiple simulations in parallel, synthetic data present themselves as an excellent alternative for AI models training and/or testing and simulated tactile data are no exception.

1.2.1 Miscellaneous Tactile Sensors Modeling Techniques

Habib, Ranatunga, Shook & Popa (2014) proposed "SkinSim", a Gazebo simulator based framework for multi-modal artificial skin. After specifying the skin parameters in the framework's user interface, like skin density, patch size and model structure, the robot placed in the simulation environment gets covered by a skin layer that consists of multi-element mass-damper-spring systems. Each skin element has a spherical geometry that minimizes the number of contacts per element and is connected to the robot's surface using a prismatic joint. Therefore, the proposed skin structure overcomes the limitation of the contact reduction schemes applied by the underlying physics engine, which is ODE (*Open dynamics engine*) in this case, and allows to simulate soft contact deformations. Underneath the skin layer, tactile sensors are placed as rectangular blocks connected with prismatic joints leading to another mass-damper-spring model with a relatively high stiffness this time. Although Gazebo has the ability to report the values for the applied contact forces, its data is noisy and inaccurate. Therefore in another work, Habib *et al.* (2014)

relied on the dynamics equations of a mass-damper-spring system to calculate the contact forces from the Gazebo reported positions and velocities of the skin elements. However, this work did not address the contact forces diffusion across the tactile elements which should be a direct result of the skin layer compliance and elasticity. Therefore, Kappassov, Corrales-Ramon & Perdereau (2020) relied on an empirical characterization procedure to simulate tactile sensing arrays. The sensors were represented in Gazebo simulation environment as single solid bodies with a triangular mesh to represent the arrays of sensing elements. Given that Gazebo is a rigid multi-body kinematics simulator and the simulated tactile sensors are usually installed under a soft dielectric and/or protective layer in real applications, a two-dimensional Gaussian-based PSF (*Point spread function*) was used to simulate the effects of the deformation of a soft layer on top of the sensor by modeling the spread of the applied contact forces throughout the sensing elements. It is worth noting that the Gaussian's standard deviation and the stiffness parameter used in the PSF were calculated empirically from the characterization of real tactile sensors using a 3 axes manual manipulation platform.

Moisio, León, Korkealaakso & Morales (2013) also created a simulation model for a tactile sensor but this time as an OpenRAVE plugin. The model took into account both the function or contact forces and positions measurement and the physical properties of the sensor like friction properties and elasticity. In their work, the contact forces were calculated using the soft contact approach that allows inter-objects penetration and taking into account the local deformation. As for the physical properties mainly related to the normal force and the tangential friction force, they were calculated by conducting a set of experiments on a real-life piezoresistive tactile sensor. It is worth mentioning that the modeled sensor was attached to a parallel jaw gripper and its performance was tested and compared to the real-life sensor. While the simulated sensor's behaviour aligns with the theoretical expectations like showing consistent results and displaying similar readings for the different taxels in the case of evenly applied external pressure however it fails to reflect the real sensor's behaviour. These deviations from the ideal simulation responses might be attributed to the real sensor's noise as well as hysteresis, creep and other factors that should be added to the simulation to minimize the gap with reality. Driess, Hennes & Toussaint (2019) introduced a foundational simulation framework, which laid the groundwork for subsequent research. In their paper, they proposed an entropy-based active tactile exploration strategy of non-convex objects using multiple end-effectors. Their approach revolved around modeling sensor-object interactions, contact mechanics, and sensor noise. The simulation setup featured the representation of objects as point clouds, synthetic contact force generation and deformations based on contact interactions. While commendable for its simplicity, the basic contact models and point-cloud representations might limit its applicability to accurately replicate complex tactile interactions or dynamic responses. However, this work contributes valuable insights into enhancing exploration efficiency, which can be relevant for generating more realistic tactile data in our simulation context.

1.2.2 FEA(*Finite element analysis*)-based Tactile Sensors Modeling

Wang, Huang, Fang, Sun & Li (2021b) introduced a method called "Elastic Interaction of Particles" for simulating the physical interaction between a tactile sensor and an object during robot manipulation. This involved a voxelization of both the sensor and the objects and a simulation of the sensor's deformation upon contact using elastic theory and numerical techniques. In this work, FEA (*Finite element analysis*) played a pivotal role, enabling accurate tactile response predictions and the simulation setup was an intricate combination of detailed object geometries, force application, and mechanical equations solving. Also, Sferrazza, Bi & D'Andrea (2020) trained a deep neural network using simulation-generated data to predict the 3D contact force distribution on the soft surface of a real-life vision-based tactile sensor. FEM (*Finite element method*) was used to handle the hyperelastic material models of the soft materials that are used to simulate the vertical indentations on the soft sensor's surface during the experiments while generating accurate force distribution data for each indentation scenario. The simulations aim to replicate real-world indentation scenarios, considering a range of depths and normal forces. These meticulous approaches demonstrated an impressive level of accuracy, rendering it suitable for tasks demanding precise force estimation and highly accurate object and shape recognition. However, the computational demands associated with FEA could hinder the real-time application of such approaches, especially in scenarios involving rapid interactions. In our project, we wanted to avoid highly computationally demanding solutions and therefore we adopted a solid bodies simulator that is relatively not computationally demanding. Similarly, Narang, Sundaralingam, Macklin, Mousavian & Fox (2021) relied on FEM to create a 3D model for the BioTac sensor using NVIDIA's GPU-based Isaac Gym simulator but this time they pushed the boundaries of tactile sensors to encompass material recognition tasks while substantially reducing the simulation time by relying on Isaac Gym's parallelization capabilities. The model was designed to predict contact forces and deformation fields for various contact interactions involving the BioTac. Similarly to the work done by Wang *et al.* (2021b) voxelization was used to represent the BioTac's external and internal surfaces to simulate the deformations and interactions accurately. The proposed FEM model uses a co-rotational linear-elastic constitutive model for internal dynamics of deformable bodies and an isotropic Coulomb contact model for interactions with external rigid objects and this model was shown to generalize well across a diverse range of objects and indentations. The developed FEM model offers an accurate and fast way to simulate contact interactions involving the BioTac tactile sensor but nontheless, it's worth noting that the simulations might still struggle to fully encapsulate the intricate tactile interactions inherent to real-world materials. While no details were presented about the specific setup of the tactile sensors, Khadivar, Yao, Gao & Billard (2023) evaluated the effectiveness of multiple algorithms for exploring the geometry of unknown objects using multi-fingered robotic hands and this included single and multi-contact exploration strategies, as well as exploration methods with dynamic hand pose adaptation and an integrated impedance controller. The algorithms were shown to adapt hand poses to object surfaces, improve kinematic properties and collect data efficiently in various scenarios, even for objects with complex shapes and this provides some insight to explore effective ways in our project to collect tactile points to represent the objects shapes efficiently through point clouds. While the proposed approach's appeal lies in its capacity to facilitate tactile simulation within realistic virtual environments and therefore enhancing algorithm development and validation, fine-tuning the simulation parameters for high-fidelity tactile data might be necessary.

1.2.3 Vision-based Tactile Sensors Modeling

With more researchers putting work into vision-based tactile sensors, several simulation models for such sensors have been developed. For instance, Agarwal, Man & Yuan (2021) introduced an optical simulation system specifically designed for vision-based tactile sensors, with a focus on the GelSight sensor. The authors used physics-based rendering and path tracing to accurately model the interaction of light with the deformable sensor surface and produce realistic images as seen by the vision-based tactile sensor. Moreover, the translucent supporting structure (gelcore) inside the sensor and the deformable elastomer surface were modeled using microfacet theory and a diffuse material model with matching reflectance properties respectively. The use of physics-based rendering for simulating tactile data is a noteworthy contribution as it allows for the creation of highly realistic synthetic datasets but one potential limitation is that the accuracy of the simulated data heavily depends on the accuracy of the physical properties and the simulation parameters used. It is worth noting that while simulating tactile sensors, which involves contact and force calculations like in our project, can also be computationally intensive however these simulations are usually less demanding than vision-based simulations as they primarily deal with physics calculations and collision detection rather than complex rendering and image processing tasks. Also, in their article, Wang, Lambeta, Chou & Calandra (2022) presented TACTO, a fast, realistic and flexible simulator that is designed specifically for vision-based tactile sensors. The software architecture of TACTO serves as a bridge between the physics simulation engine and the rendering engine, by preloading gel surface and object meshes into an OpenGL scene, synchronizing their poses with the physics simulation and efficiently generating depth and RGB images for tactile simulations. Moreover TACTO's flexibility lies in its ability to simulate contact forces and deformations using a user-defined deformation function and to being calibrated using real-world sensor data for more realistic simulations as well as in its adaptability to different sensor designs by allowing easy changes to configuration files. The introduction of TACTO offers a significant contribution by providing a comprehensive, open-source simulator for high-resolution vision-based tactile sensors. The focus on real-time interactions and GPU acceleration enhances the efficiency of data generation and therefore allowing researchers to create large-scale synthetic datasets rapidly. However, while TACTO addresses the need for fast and flexible simulations, the trade-off between simulation speed and physical accuracy could potentially affect the realism of the generated tactile data. Moreover, Si & Yuan (2022) presented "Taxim," a simulation model tailored specifically for GelSight tactile sensors with the focus on by combining optical and marker motion field simulations to replicate the behavior of GelSight sensors in a virtual environment and therefore facilitating data generation and analysis for tactile perception research. The simulation setup involves replicating the GelSight sensor's behavior through an example-based approach. In other words, real-life GelSight measurements are used as input and the model is designed to predict the real sensor's response to various deformations and interactions with objects with a minimum sim-to-real gap. The development of an example-based simulation model like Taxim is particularly valuable for replicating the complex behavior of GelSight tactile sensors but its applicability to other types of tactile sensors may be limited. Additionally, the accuracy of the simulation heavily relies on the quality and diversity of the example-based dataset, potentially leading to challenges in accurately representing the sensor's behavior in all scenarios. Finally, Dikhale *et al.* (2022) combined visual and tactile data to estimate the 6D pose of in-hand objects during a robotic gripper's interaction with its external environment. The simulation aspect of the paper focuses on generating a synthetic dataset for training a machine learning network. The simulation setup was built in Unreal Engine 4 and consisted of a gripper mounted on a robotic arm and equipped with tactile sensors, and virtual cameras attached to each finger's phalanx. The mounted tactile sensors were simulating pressure-based tactile sensors and the authors used forward kinematics to keep track of the 3D positions of their taxels (tactile sensing elements) that enter in contact with the object's surface. As for the virtual cameras, these were simulating optical-based tactile sensors like Gelsight sensors and they were used to capture color and depth images when the gripper makes contact with the object in the simulation environment. Moreover, the authors applied domain randomization to the simulation environment by varying the backgrounds, floor surfaces as well as lighting and other factors to increase the generalizability of the the machine learning network to be trained and make it more adaptable to real-world conditions. While similarly to our project the activated taxels positions were tracked for point cloud generation, the proposed approach's addition of optical-based simulated sensors showcased its accuracy, particularly under occlusion and highlighted the potential of multi-sensory fusion. However, the method's robustness and adaptability across various interaction scenarios requires further exploration.

1.3 Tactile Sensing for Object or Shape Recognition

Tactile sensing for object recognition represents a groundbreaking fusion of human-inspired perception and cutting-edge robotics technology. By leveraging the sense of touch, robots equipped with tactile sensors can gather rich and detailed information about the physical characteristics of objects they interact with. This includes data on an object's shape, size, texture and rigidity. These tactile cues provide valuable insights that complement traditional visual perception methods, making it possible to recognize and identify shapes, and sometimes even objects with a higher degree of accuracy, especially in scenarios where visual cues may be limited or unreliable. Many researchers have leveraged the information provided by tactile data for object and shape identification. For instance, Zhang, Atanasov & Daniilidis (2017) used contact coordinates based point clouds to classify the objects grasped by a robotic hand. The objective of the project was to minimize both the object misclassification probability and the number of needed grasps. The object and shape recognition was formulated as a MDP (*Markov decision process*) where the optimal grasping sequence was approximated by Monte Carlo tree search. Zhang, Kennedy, Hsieh & Daniilidis (2016) presented a haptic descriptor for 3D objects classification that is invariant to the object pose or motion. After a few probing touches of an object, the recorded contact positions are sampled as sets of three points on the surface of the grasped object and the resulting triangular mesh is used to generate a 3D histogram that describes this object. In order to build the previously mentioned histogram descriptor, three parameters that uniquely define a triangle are selected (i.e. a combination of angles and side lengths) and the distribution for each parameter is presented along an axis with a specific number of bins. These robust descriptors were then fed into a SVM (*Support vector machine*) and a FFNN (*Feed forward neural network*) to associate different objects with specific triangular parameters-based 3D histograms.

In our project, we focused in the identification of only primitive shapes. And unlike the previously described methodologies, we did not use AI algorithms for the identification process but instead we applied a matching between the tactile point clouds generated by our sensors while grasping the shapes and CAD (*Computer-aided design*) models of these shapes and we have finally relied on statistical distance metrics to assess the quality of each match.

1.4 Simulator Selection

As previously discussed in the introduction, simulation was a major aspect in our project and as such, a literature review was conducted to compare the different simulators used in the robotics field nowadays and therefore make an informed selection of a simulator that meets our requirements for this project. Given that ROS (*Robot operating system*) is a widely adopted robotics cross-platform firmware, and many researchers from our laboratories have previous experience with it including me and my supervisor, our review exclusively covered the simulators that provided a ROS compatibility in one way or another. For instance Farley, Wang & Marshall (2022) compared four of the most widely used simulators in the robotics field: CoppeliaSim or formerly known as V-REP, Gazebo, MORSE and Webots. The investigated simulators were compared both qualitatively and quantitatively. On the qualitative level, the simulators were compared based on the key attributes presented in table 1.1.

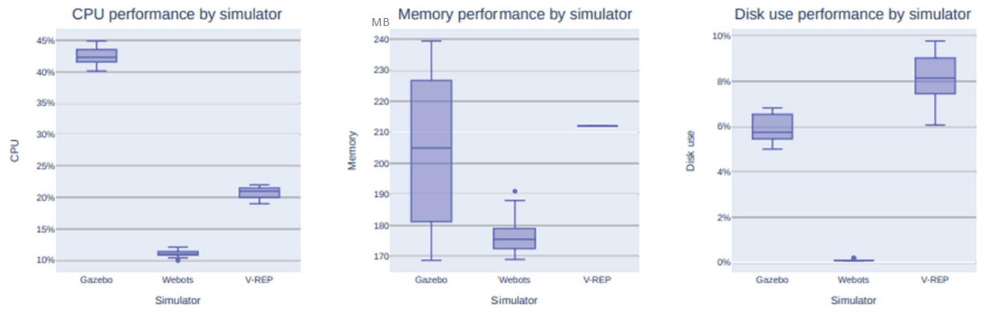
While CoppeliaSim and Gazebo support multiple physics engines including Bullet, ODE among others and multiple programming languages including C++ and Python on top of offering a plethora of simulation assets including models for mobile robots and environments, each of MORSE and Webots are relatively limited in terms of physics engines support and simulation models libraries. As for the quantitative level the simulators were compared relative to a ground truth real life experiments after creating an adequate virtual world in each. More specifically IMU (*Inertial measurement unit*) data were collected from a real Husky A200 robot driving on mixed terrains and compared to the synthetic data generated by each simulator. As a result, Gazebo displayed a poor CPU (*Central processing unit*) efficiency but it showed excellent IMU data accuracy levels.

Table 1.1 Simulators qualitative features evaluation
Retrieved from Farley, Wang & Marshall (2022)

Metric name	CoppeliaSim	Gazebo	MORSE	Webots
Free to use	True	True	True	True
Open source	False	True	True	True
ROS compatibility	A built-in plugin provided	out of the box	Out of the box	A built-in plugin provided
Programming languages	C/C++, Python, Lua, MATLAB, Java, Octave	C/C++, Python	Python	C/C++, Python, Java, MATLAB
UI functionality	Full functionality	Full functionality	Visualization only	Full functionality
Model format support	URDF, SDF, Stl, Obj, Dxfm Collada	URDF, SDF, Stl, Obj, Collada	Blend	Proto Nodes
Physics engine support	Bullet, ODE, Vortex, Newton	Bullet, ODE, DART, Simbody	Bullet	ODE

Moreover, Ayala *et al.* (2020) have conducted a quantitative comparison among Gazebo, Webots and CoppeliaSim. Each of the three simulators have executed 20 times the same scenario during which a NAO robot must navigate an obstacles filled environment to reach a specific goal position. During these experiments, the CPU, memory footprint and disk access were monitored. The experiments results are shown in figure 1.1.

It seems that Webots required the least processing power with an average of 11.05% for CPU usage, followed by Coppeliasim or V-REP with around twice of CPU usage. Gazebo required the highest amount of CPU usage, with an average of 42.38%. As for the memory footprint Coppeliasim presented the most stable amount of required memory as opposed to Gazebo where the variation in the memory requirements varied significantly among the different executions. Finally, Coppeliasim required the highest amount of disk access with an average of 8.16%, followed by Gazebo with 5.96% and finally Webots with 0.12%. Therefore, Webots has excelled, requiring the least amount of computational resources to perform the simulation task.



(a) CPU use boxplot of 20 executions. The simulator with smallest standard deviation is Webots with 0.56, followed by V-REP with 1.04, and Gazebo with 1.46.

(b) Memory boxplot of 20 executions. The simulator with smallest standard deviation is V-REP with 0.0, followed by Webots with 6.36, and Gazebo with 24.0.

(c) Disk access boxplot overall of 20 executions. The simulator with smallest standard deviation is Webots with 0.04, followed by Gazebo with 0.61, and V-REP with 1.01.

Figure 1.1 Performance results for the different simulators after executing the same scenario for 20 times
Retrieved from Ayala et. al. (2020)

Although other robotics simulators were shown to be superior like V-REP and Webots, Gazebo's descent performance, its open source nature, native ROS compatibility encouraged its selection among other simulators for this project. Moreover, its large library of models reflect how active is the community that stands behind Gazebo which goes a long way for our project in terms of support and continuous improvement. Finally, the previous expertise that specifically I and my supervisor had with Gazebo, has fortified further our adoption of this simulator for the project.

CHAPTER 2

TACTILE SENSOR: THEORY AND FUNDAMENTAL PRINCIPLES

2.1 Introduction

The aim of this chapter is to present the proposed capacitive tactile sensor that completely covers the Allegro anthropomorphic robotic hand. First, we go over the physics theory behind the sensor's operation, then we explain the employed mutual capacitance sensing technology before finally presenting the sensor's characterization. As previously mentioned in the literature review the presented sensor is based on the previous work done by Bao group at Stanford university.

2.2 General Principle

The developed tactile sensor in this project was based on the capacitive technology. As previously discussed in the literature review section, this technology was selected among many because it offers an excellent trade-off between resolution, precision, measurement range, responsiveness and design flexibility. Moreover, our developed sensor has a relatively simple design, allowing it to easily and comprehensively cover an anthropomorphic robotic hand and in addition to measuring contact forces, it can be used for proximity sensing.

The presented sensor is designed to measure the static forces applied on the surface of the allegro hand. As previously mentioned in the literature review, the applied forces are reflected as capacitance changes which are translated in our design into digital readings by relying on the electrical properties of capacitive objects. For instance, a votlage difference V applied to an object with capacitance C will create an electric field with charge Q as per equation 2.1.

$$Q = CV \tag{2.1}$$

As it will be explained in a subsequent section, an external force applied to the sensor will disturb the established electrical field by changing the dielectric permittivity of the medium.

2.3 Fringe Effect and Interdigitated Design

In order to efficiently interact and navigate with their environment, humans rely on the simultaneous inputs from multiple sensors. Similarly, robotics applications could greatly benefit from multiple sensory inputs. More specifically, providing both pressure and proximity sensing capabilities allows to develop robotic systems that can interact autonomously and safely in cluttered environments. For instance, robots that can only detect pressure contact cannot decrease their speed while approaching fragile objects and therefore risk to accidentally destroy them. Furthermore, previous research has shown the importance of the robotic distinction between materials for a better autonomous interaction with the surroundings (Kirchner, Hordern, Liu & Dissanayake, 2008), (Alagi, Heiligl, Navarro, Kroegerl & Hein, 2018) and (Ding, Zhang & Thomas, 2018).

Therefore, it would be very useful to equip the robots with a multitude of sensory inputs. However, this often requires the use of multiple sensors that occupy more space and therefore reduce the density of sensors that can be integrated on a robotic surface and therefore limiting the sensory resolution. Alternatively and in order to overcome the multi-sensory input challenge the fringe fields in capacitive sensors have been lately used to design sensors with both contact pressure and non-contact proximity sensing abilities like in the work presented by Ruth *et al.* (2021).

A capacitor is typically composed of two electrodes that are separated by a dielectric. When voltage difference is applied to these electrodes, an electrical field appears not only between the electrodes but it also extends to a certain distance beyond that and therefore giving what is called fringe fields or fringing effect. The concept is depicted in image 2.1.

Incoming objects create a disturbance in the fringe field and therefore can be detected by the fringe field sensors. However, the proximity sensing capabilities introduce negative impacts on the contact pressure sensing sensitivity, which can be problematic especially when it becomes challenging to distinguish between contact and noncontact signals for a low pressure sensitivity. Therefore the work done by Ruth *et al.* (2021) and previously discussed in the literature review section, allowed to create a sensor that has a strong performance in both pressure and proximity

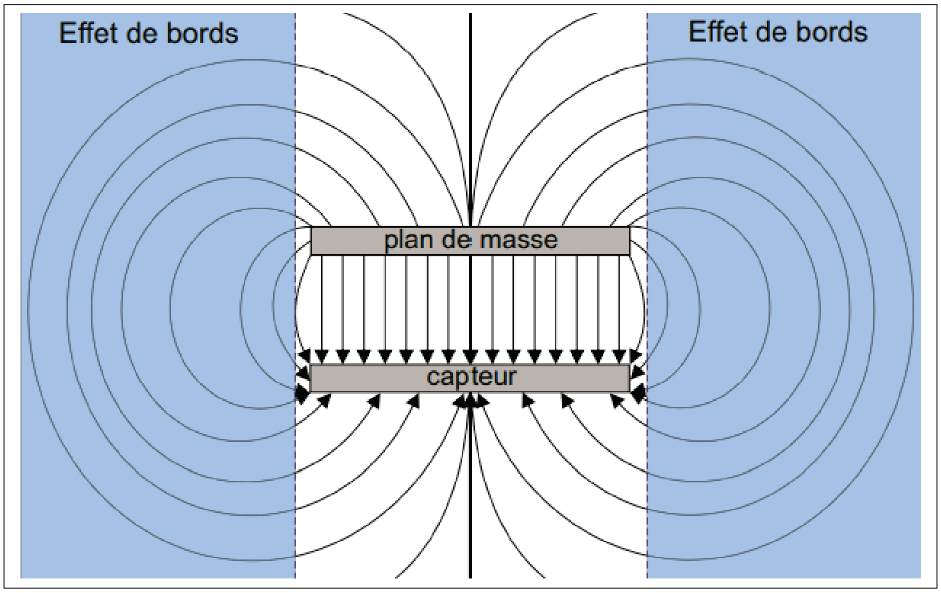


Figure 2.1 Fringe effect: The drawn lines show the extents of the electrical field that is created by a parallel plates capacitor Retrieved from Maslyczyk et. al. (2016)

sensing by relying on the fringe effect and the interdigitated electrode design. The interdigitated design adopted in our project and inspired by the work of Ruth *et al.* (2021) is presented in figure 2.2. On the left side the taxel, which is a portmanteau for tactile pixels and the sensory unit for the developed tactile sensor, illustrates the interdigitated electrodes layout. These electrodes are connected to an electric signal TX (*Transmitter*) line and a RX (*Receiver*) line that are crucial for sensory readings and will be discussed in the subsequent section. As for the right side the sensor setup is presented. In this setup, a silicone-based dielectric with a surface featuring microstructural cylindrical heads is placed on top of the electrodes and therefore directly interfering with the fringe field.

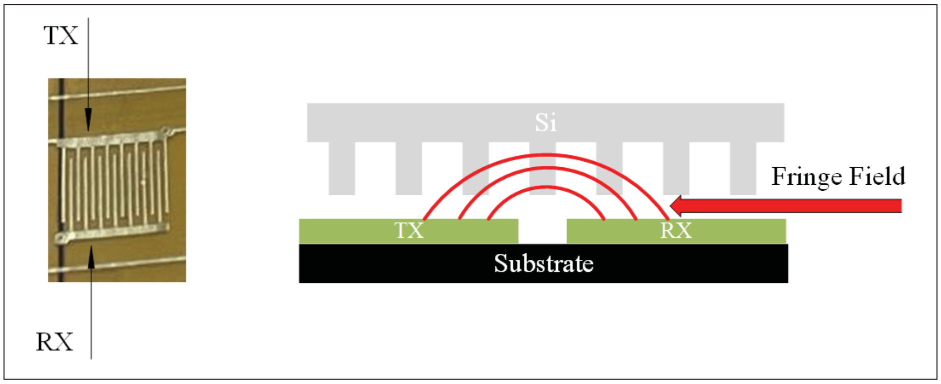


Figure 2.2 Interdigitated capacitive tactile sensor: The interdigitation amplifies the electrical field between the TX and the RX lines located under the Silicone-based dielectric layer

Given that the area of the overlapping electrode surfaces as well as the distances among these electrodes are constants, the variation of the permittivity of the electric field medium is the only parameter that controls the variation in the capacitance of the developed tactile sensors. For instance, an approaching object disrupts the fringe field in the air above the sensor and therefore can be detected by the triggered capacitance variation even without direct contact whatsoever. Moreover, if such an object makes contact with the sensor, the dielectric layer will be compressed and the air in the gaps between the cylindrical protrusions will be displaced. Given that the dielectric constant of the air is lower than that of the dielectric material, this compression will lead to an increase in the effective dielectric constant and therefore a higher capacitance value.

Interdigitating the electrodes in a capacitive sensor allows to increase the number of electrode pairs within a given area, which leads to an enhanced electric field strength that is reflected in a higher sensitivity for both contact and non-contact modes. This was the driving reason for adopting such a design over other alternative models of fringe field sensors for this project.

2.4 Mutual-Capacitance Sensing Technology

In order to monitor the capacitance value of the developed tactile sensors, we used a PSoC (*Programmable system-on-chip*) 4200 microcontroller from Cypress and we adopted the CSX

(*Capsense crosspoint*) mutual capacitance tactile sensing method. This Infineon patented technology is well known for its best-in-class SNR(*Signal to noise ratio*) (AG, 2022).
In our project, a total of 17 sensors were developed based to cover an Allegro hand. The sensors can be grouped into seven different categories according to the part of the robotic hand that they are covering, as shown in table 2.1.

Table 2.1 The different types of the developed tactile sensors and their characteristics

Sensor's hand part	**Total taxels**	**Total RX**	**Total TX**
Back of hand	118	12	11
Fingertip	66	9	13
Medial back	30	8	8
Medial front	27	4	8
Palm	121	11	13
Proximal back	78	11	11
Proximal front	65	11	10

Each tactile sensor type has a specific number of taxels which is dictated by the surface area available for each of the allegro hand's part. These taxels are arrayed into matrices of different sizes where each taxel is the formed by the interdigitation of a TX and a RX electrodes as previously shown in figure 2.2. More specifically, all the taxels lying on the same row or column in a matrix, share the same electrode which is connected to either a TX or a RX line on the PSoC. This electrodes' layout is depicted in figure 2.3.

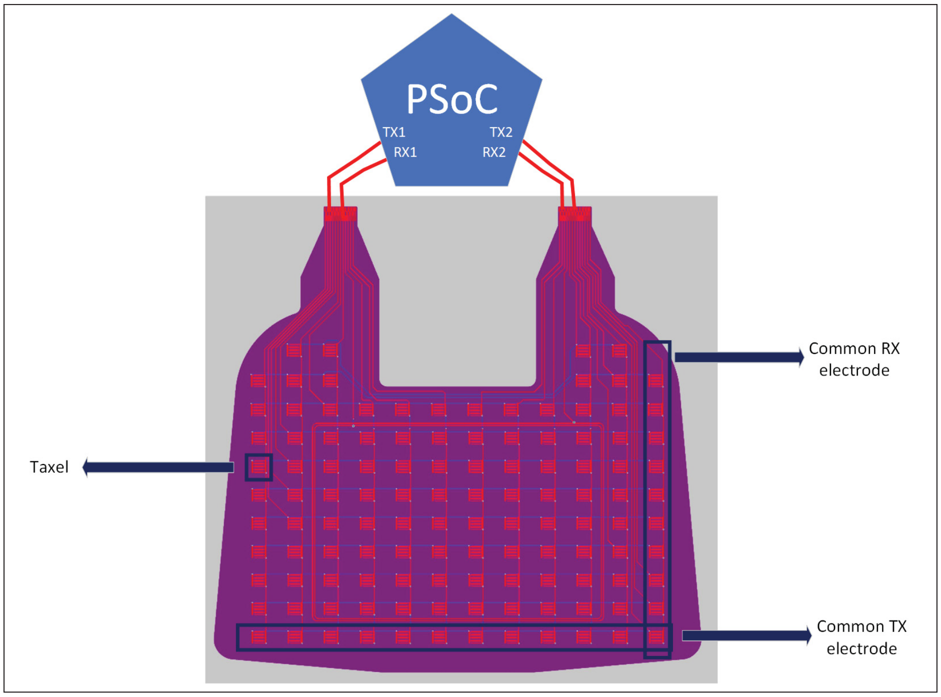


Figure 2.3 Electrodes layout and connectivity for the back of hand tactile sensor

Using the previously mentioned CSX tactile sensing method, the capacitance between two electrodes, a TX and a RX is measured using electrical pulses. In more details, a a square wave digital voltage signal switching between high and ground is sent to the active TX electrode. Given the previously discussed electrodes layout, this TX line basically forms a series of capacitors with the neighbouring RX lines. Therefore, when the voltage signal is high, the active TX line is charged through the microcontroller and when the voltage signal becomes low, the electrical charge is transmitted to the corresponding RX lines which are then individually and sequentially fed into a multiplexer and therefore each RX electrode gets discharged, generating a series of currents that are subsequently fed into a current to digital converter to give digital readings called raw counts that are directly proportional to the C_M (*Mutual capacitance*) between the corresponding two electrodes as per equation 2.2.

$$Rawcount = G_{CM}C_M \tag{2.2}$$

where G_{CM} is the capacitance to digital conversion gain.
The same procedure is applied to the next TX line and so on, until a complete sensor cycle iterates though all the taxels of a single sensor. At this stage the developed PSoC firmware goes over the stored raw counts for each taxel before communicating them to a computer. It is worth noting that this whole process was described for a single sensor and the different sensors were actually connected through a network with a specific communication protocol to ensure a smooth and reliable sensory data transfer and this work will be detailed in the next chapter.

The complete measurement process for a single sensor is depicted in figure 2.4.

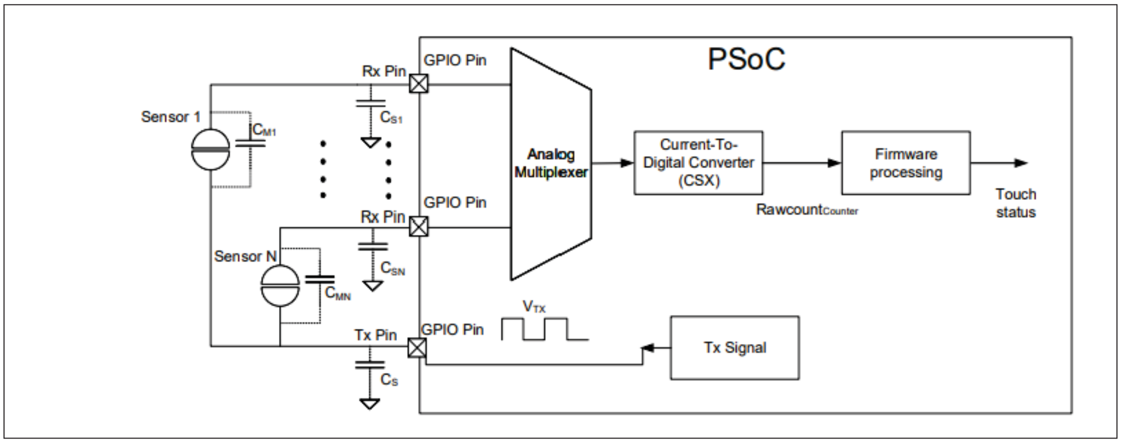


Figure 2.4 Simplified diagram of the CSX mutual capacitance sensing method: The PSoC microcontroller transmits electrical signals through the TX lines and receives a signal response though the RX lines to analyze the capacitance variation at each taxel

2.5 Sensor Characterization

Each of the previously discussed sensor types was printed on a FPCB with 3x3 mm taxels that was connected to a corresponding PCB on which a dedicated PSoC microcontroller was integrated and covered with a Silicone-based dielectric. While the design will be further discussed in details in the next chapter, this section is dedicated to better understand the behaviour and performance of the developed sensor. As part of our project, experiments were conducted on the back of hand sensor whose layout was previously shown in figure 2.3. A probe with a 12 mm diameter was used to progressively load and unload the sensor up to 50 N. The sensor's response was recorded and is shown in figure 2.5.

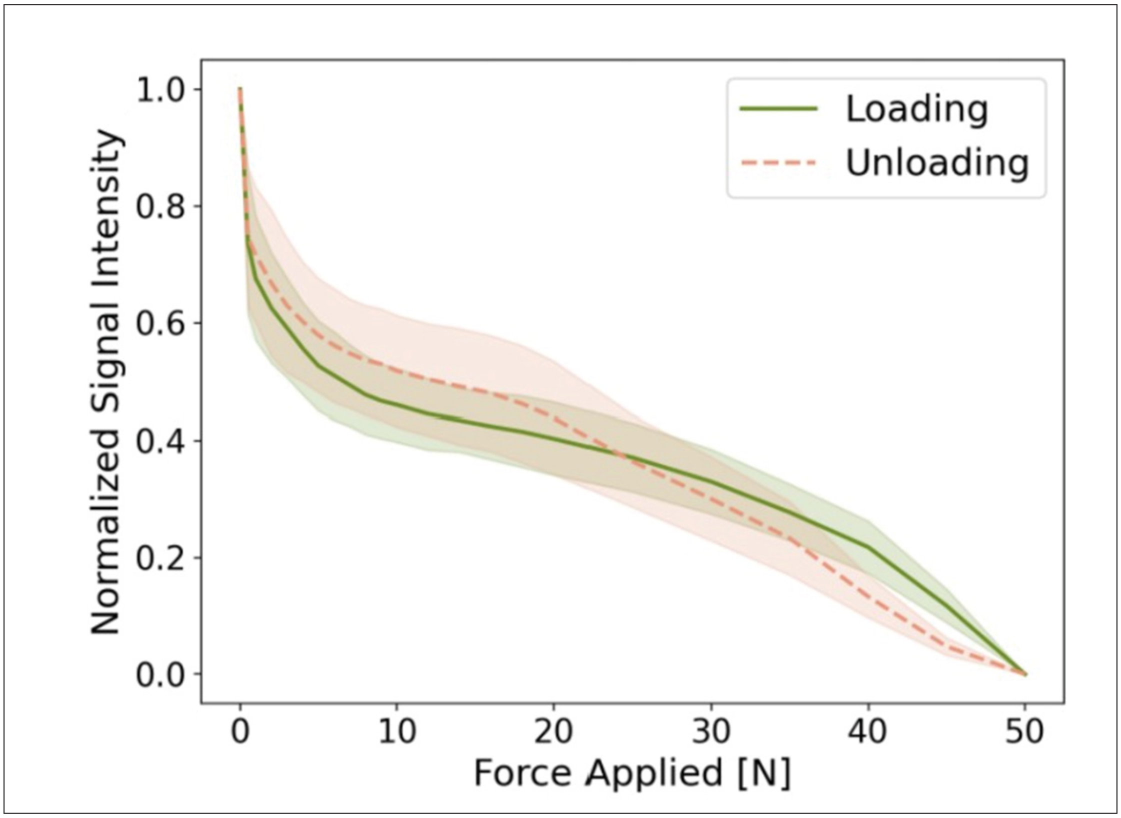


Figure 2.5 Signal variation with respect to the force applied to the back of hand sensor: The mean signal of 10 taxels during loading and unloading are shown by the solid and dashed lines respectively. The standard deviation is represented by the shaded regions

Each taxel provides a different constant digital raw count reading at the initial state with no externally applied force and it seems obvious that as applied force increases, the raw count decreases until reaching a saturation level that was spotted at 50 N in this case. It is worth noting that each taxel's reading was normalized relative to its corresponding initial unloaded value to standardize the responses of the different taxels, ensuring that their readings fall within the same range between 0 and 1. Moreover, using a 1.9 mm diameter probe we verified that the sensor can detect point contact forces as light as 0.06 N.

Finally, it is worth mentioning that given the small dimensions of the designed taxels, the developed electrical fields at the electrodes interdigitation were not powerful enough to allow the sensor to effectively display its proximity sensing capabilities. To address this issue, additional larger electrodes pathways can be added to the current sensor's layout in future design iterations. However, this capability was not needed in this project and therefore no additional work was done in this area.

CHAPTER 3

SENSOR REAL LIFE SETUP

3.1 Introduction

In order to validate the functionality of the tactile sensor that was previously discussed in chapter 2, a hardware setup was created to build an interconnected network of tactile sensors and mount it on a robotic manipulator. In this chapter, we go over the design of the network and the mounting mechanism and finally the robotic manipulator's control and the sensory data handling are discussed.

3.2 Sensory Network Design

Given the 16 DoF of the used Allegro hand, it was decided to design and equip each part of it with a dedicated sensor in order to preserve its dexterity and maintain reliable sensory readings. Towards this goal and as previously described in chapter 2, a total of 17 sensors with 7 different designs were created to cover as much as possible of the hand's surface. All the mounted sensors are shown in figures 3.1 and 3.2.

In view of the distinctive taxels count and layout inherent to each sensor design, which is intricately shaped by its spatial dimensions, influenced by its placement on the allegro hand, it is noteworthy that all sensors collectively adhere to a common foundation of design principles and manufacturing methodologies that will be thoroughly discussed throughout this chapter. Figure 3.3 shows the different components of the back of hand sensor.

In accordance with the previous discussion in chapter 2, a dedicated PSoC microcontroller was employed with the purpose of monitoring the capacitance variation for each sensor. Each PSoC unit was seamlessly integrated into a distinct PCB, alongside additional electronic components to ensure an uninterrupted flow of capacitance data from each sensor. The design of these PCBs was meticulously executed within the Altium PCB design software, adhering to the

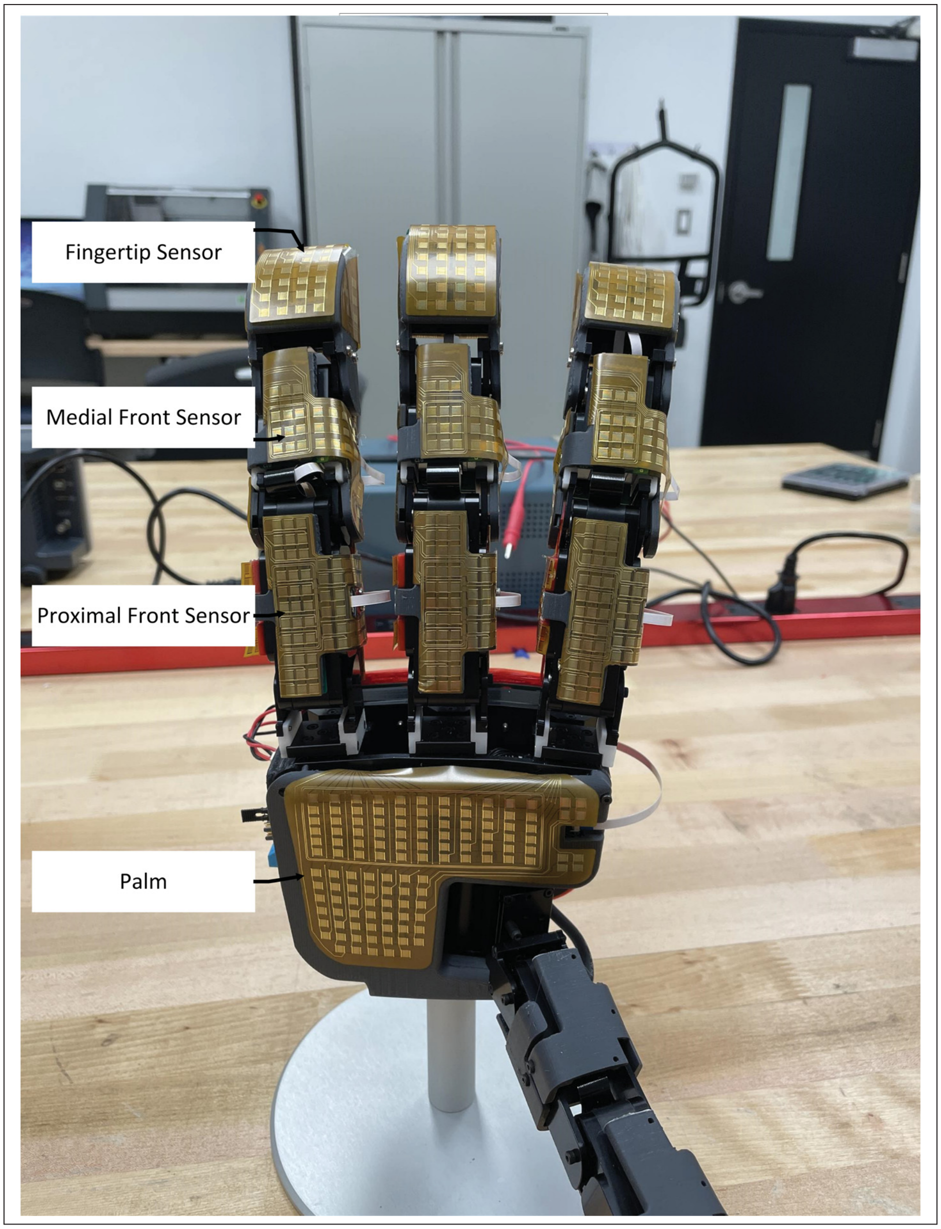


Figure 3.1 Allegro Hand With Mounted Front Sensors

guidelines provided by Cypress Semiconductor company. The design process encompassed strategic optimization of the placement of the electronic components, enabling appropriate power supply, mitigating sensory signal noise levels, and establishing efficient pathways for signal communication. Figure 3.4 shows the first version of the Altium PCB design (on the left

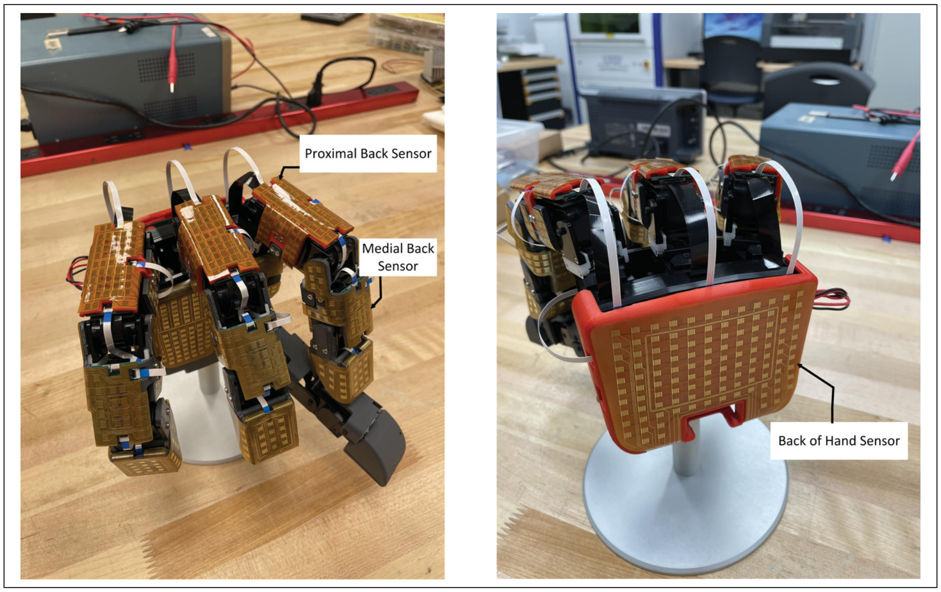


Figure 3.2 Allegro Hand With Mounted Back Sensors

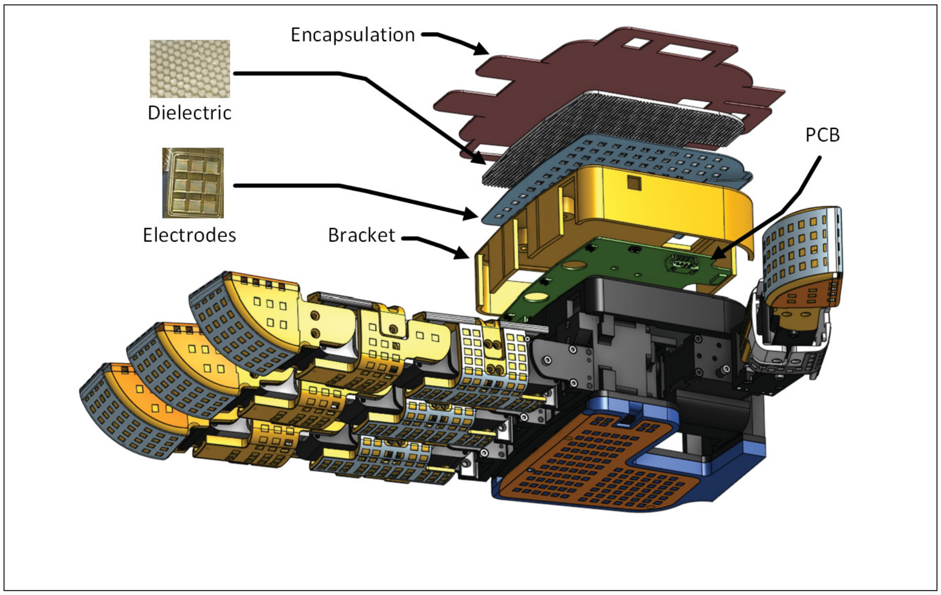


Figure 3.3 Back of Hand Sensor Components

side) and the final version of the actual PCB after component soldering (on the right side) of the medial back sensor.

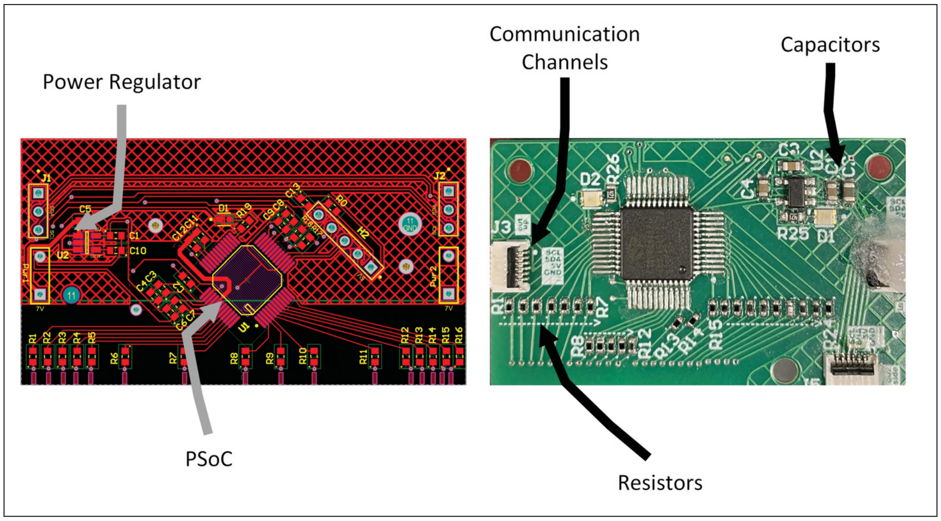


Figure 3.4 Medial back PCB design

The tactile sensors were also designed in Altium software but printed on FPCBs. More specifically, the RX and TX electrodes that were previously discussed in chapter 2 were printed on a polyamide substrate to create a flexible sensor that conforms to its installation surface and can be easily connected to the previously discussed rigid PCBs. Figure 3.5 shows the design for the palm sensor.

The PCBs were securely mounted on the allegro hand, alongside the associated sensors, utilizing specifically designed plastic brackets manufactured in the CoRo lab. The 3D printed components were meticulously designed to ensure a sufficient space for accommodating the PCBs, their connection cables, and the tactile sensors while ensuring no interference with the natural motion of the allegro hand's joints during the grasping process. The 3D printed brackets are illustrated in figure 3.6, where the palm, back of hand and central hub brackets are shown on the left side and the finger brackets are show on the right side.

Finally, a Silicone-based mictostructured dielectric was encapsulated on top of the FPCB layer in order to translate any external contact pressure into a permittivity and therefore a capacitance change as previously discussed in chapter 2. The procedure for preparing the dielectric for each sensor is depicted in figure 3.7.

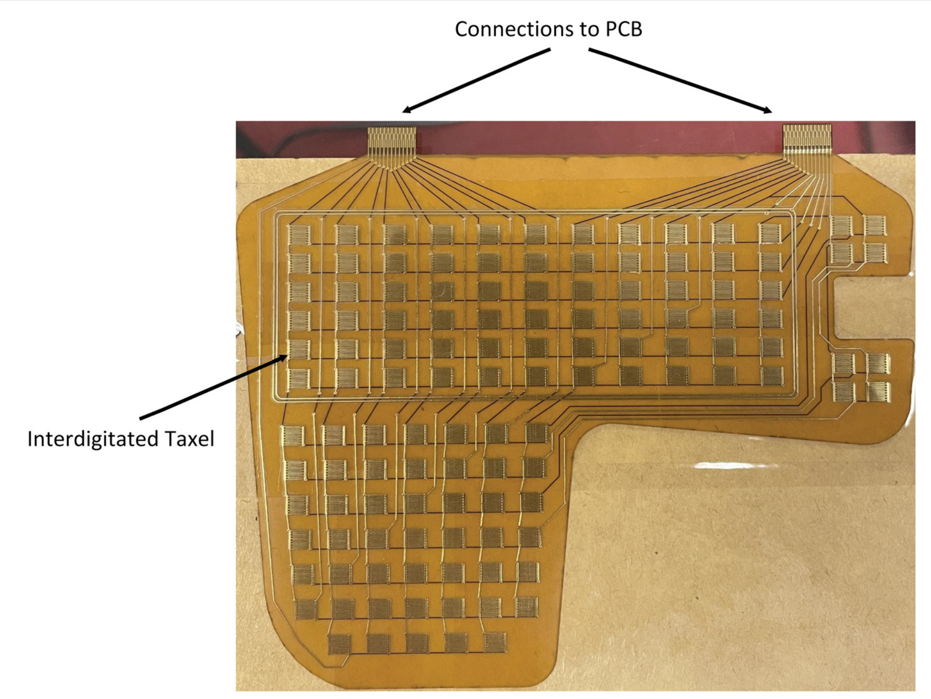


Figure 3.5 Palm FPCB Design: The sensor's electrodes were printed on a flexible paper-like substrate that can be directly connected to a rigid PCB with an onboard PSoC for sensory data processing

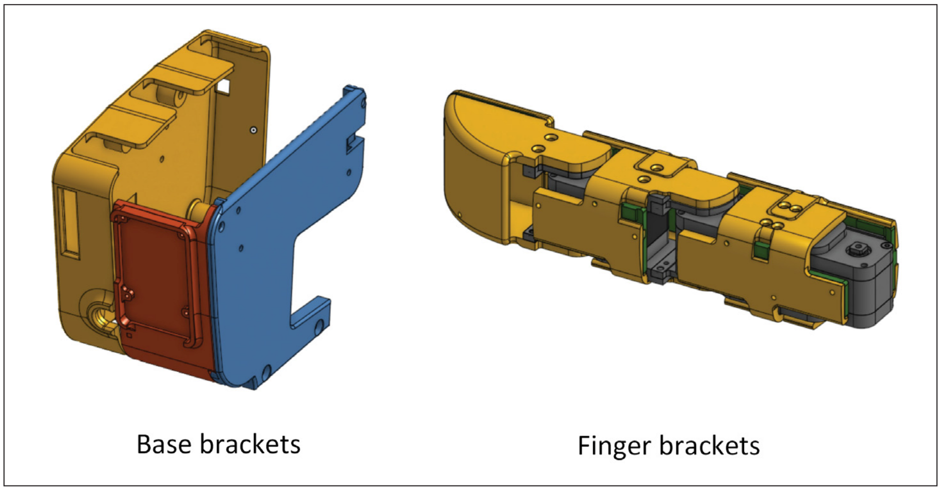


Figure 3.6 Allegro Hand Brackets: Used to mount the tactile sensors and their corresponding PCBs on the Allegro hand

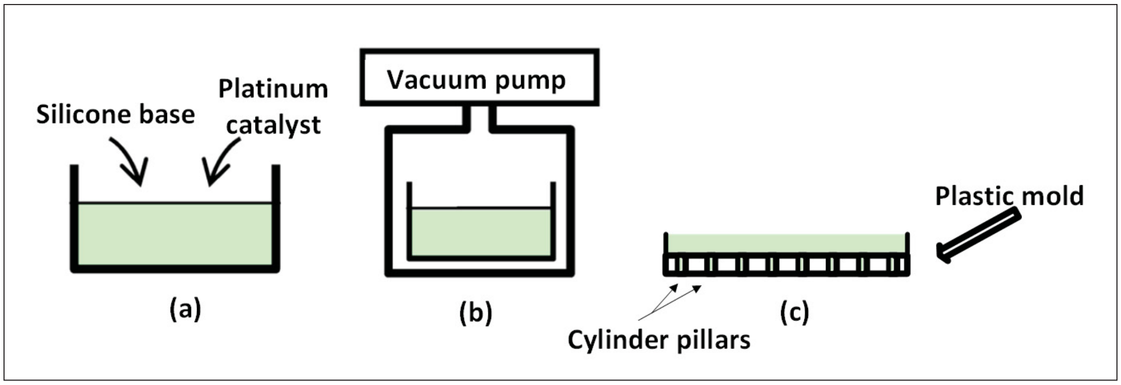
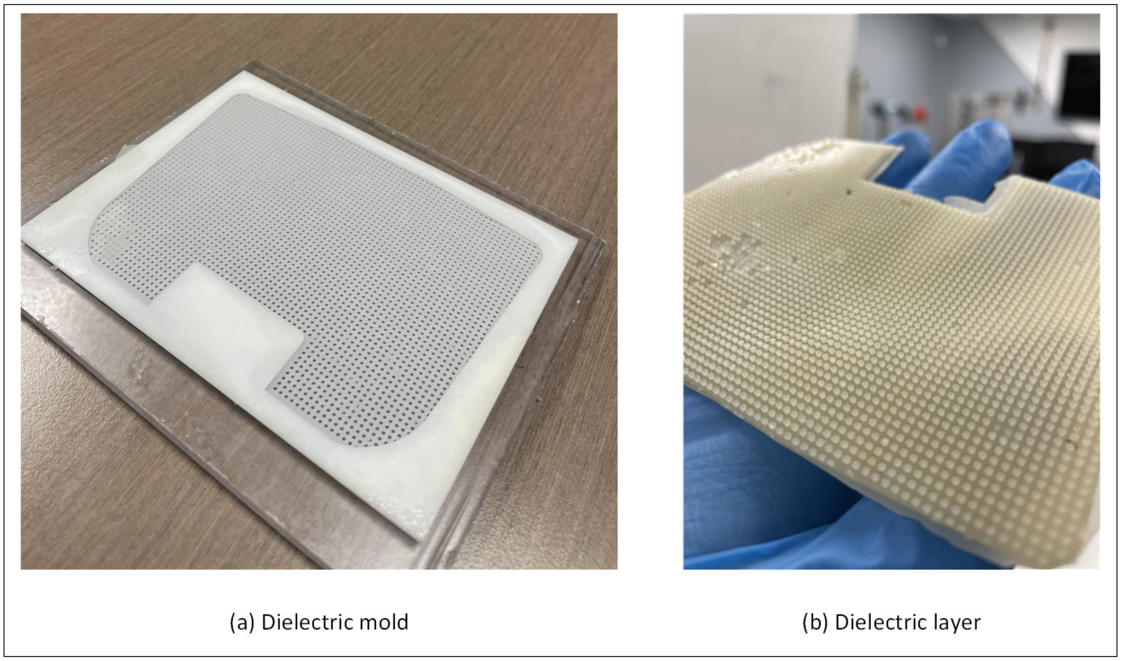


Figure 3.7 Dielectric fabrication procedure: **(a)** Silicone is mixed with platinum to form the Ecoflex-30,**(b)** The dielectric liquid is degazified in a vaccum container to prevent any trapped air in the final dielectric layer, **(c)** The dielectric liquid is cast in a plastic mold that was engraved using a laser cutter

Ecoflex-30 was used to prepare the dielectric layer of the sensor. This material is a platinum-cured or platinum-catalyzed silicone that was prepared by first mixing the silicone with the platinum catalyst in a container. Then, the mixture was degasified using a pump and finally the dielectric was cast in plastic molds that were created using the laser cutter available in the CoRo laboratory for the different sensor types. The mold for the back of hand sensor and a close up view of the corresponding dielectric layer are shown in figure 3.8

The textured dielectric's surface features 0.8 mm diameter and 0.8 mm height cylindrical pillars with gaps in between so that when compressed, the dielectric constant of this layer changes which leads to semantic sensory readings as previously discussed in chapter 2.

Ecoflex-30 is known for a multitude of attractive properties that guided its selection as a dielectric material in our project. For instance, this material has a relatively low dielectric constant which helps minimizing the capacitive crosstalk among the tactile sensor's taxels. It is worth mentioning that this dielectric constant is still higher than that of air, which is an essential criteria for our sensor's functionality that is based on the relative permittivity change of the medium upon contact. Furthermore, this silicon type is elastic and flexible and therefore can conforms well with the different surfaces of the Allegro hand. Moreover, platinum-catalyzed silicons are known for their low compression set, which means that they can recover their


(a) Dielectric mold

(b) Dielectric layer

Figure 3.8 Dielectric mold used for casting and dielectric layer showing the cylindrical pillars on the surface

initial properties and shape after being compressed. This property makes the Ecoflex material ideal for our sensor application since this will facilitate the restoration of the aforementioned cylindrical pillars to their original states. Therefore, this enables the sensor to consistently yield closely similar initial readings, mitigating the impact of compression and thereby enhancing the repeatability of the sensory data. Moreover, the Ecoflex-30 is suitable for casting since it releases easily from molds and this is crucial to avoid the destruction of the cylindrical microstructures on the surface of our cast dielectric.

One of the main objectives is to achieve a comprehensive sensory coverage of the allegro hand by integrating multiple live streaming tactile sensors. To ensure efficient connectivity, flexible flat cables were utilized to accommodate both power and communication lines for interconnecting the sensors.

The chosen wiring scheme for this sensory network was a star configuration, wherein the principal PCB serves as the central hub and is connected to the back of hand sensor from which multiple branches are extended towards the palm and each of the allegro hand fingertips and therefore facilitating a well-organized and effective system architecture. This wiring configuration was selected for its several advantages. Firstly, it allows for a better organization and management of the electrical connections since each branch originates from a central point which simplifies troubleshooting and maintenance as individual branches can be easily isolated and tested. Secondly, the star configuration reduces the impact of a single point of failure. In other words, if one branch or device encounters an issue, the other branches remain unaffected, promoting overall system reliability.

The sensory network was powered with an external DC power supply was connected to the central hub PCB, delivering a 5.7 volts. The same voltage was delivered to each of the network's PCBs through the previously mentioned flat cables and each PCB was equipped with a linear voltage regulator IC (*Integrated circuit*) on board to deliver 3.3 volts to the correspondent PSoC. The reason behind this design is to avoid any fluctuations in the electrical power delivered to the PSoCs.

As for the inter-sensory communication, the I2C (*Inter-integrated circuit*) protocol was adopted for several reasons. One of the key features of I2C is its simplicity and ease of implementation. The protocol requires minimal hardware: pull-up resistors on the SCL (*Serial clock*) and SDA (*Serial data*) lines. Another I2C feature, is its built-in bus arbitration and collision detection mechanisms, which means if multiple devices attempt to communicate simultaneously, the protocol automatically resolves the conflicts, ensuring reliable and efficient data transmission. Moreover, the I2C protocol enables easy interfacing and control of multiple devices on the same bus. Particularly in this project, each sensor connected to the I2C bus has a unique address that allows for individual sensor selection and communication. Figure 3.9 shows the hexadecimal addresses of all the sensors that were mounted on the Allegro hand.

Within this project, the PCB of each sensor is designated as an I2C slave, while the central hub PCB functions as the I2C master. Communication initiation on the I2C bus occurs when the master sends a start condition, asserting the SDA line while maintaining a high SCL line and thereby signifying the start of a communication sequence. The master transmits the address of the intended slave device, referencing one of the addresses previously depicted in Figure 3.9.

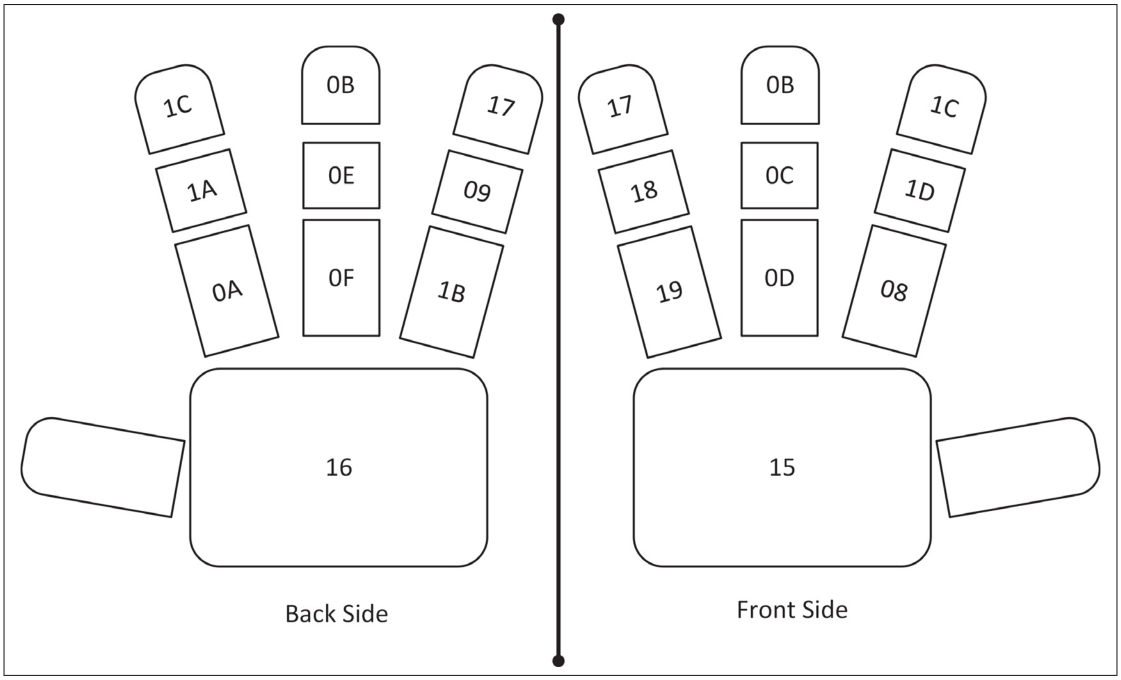


Figure 3.9 I2C slaves addresses used for intercommunication among the tactile sensors mounted on the Allegro hand

Following the transmission of the slave device's address, the master releases the SDA line and awaits an acknowledgment from the addressed slave device.

Upon successful address acknowledgment, the slave proceeds to transmit a tactile message to the master. Each slave's tactile message encompasses the raw counts of the corresponding sensor's taxels, which were discussed in Chapter 2, accompanied by the corresponding timestamp and sensor address. Subsequently, this message is directed to a USB (*Universal serial bus*) to Full Handshake UART (*Universal asynchronous receiver-transmitter*) IC integrated on the central hub PCB. This configuration facilitates the final transmission of the message via a USB cable to a computer, where it is published on a dedicated ROS topic specific to that particular sensor. The sensory data was constantly acquired and published on the corresponding ROS topics at a rate of six Hz(*Hertz*). The parsing process of the sensory data is illustrated in figure 3.10.

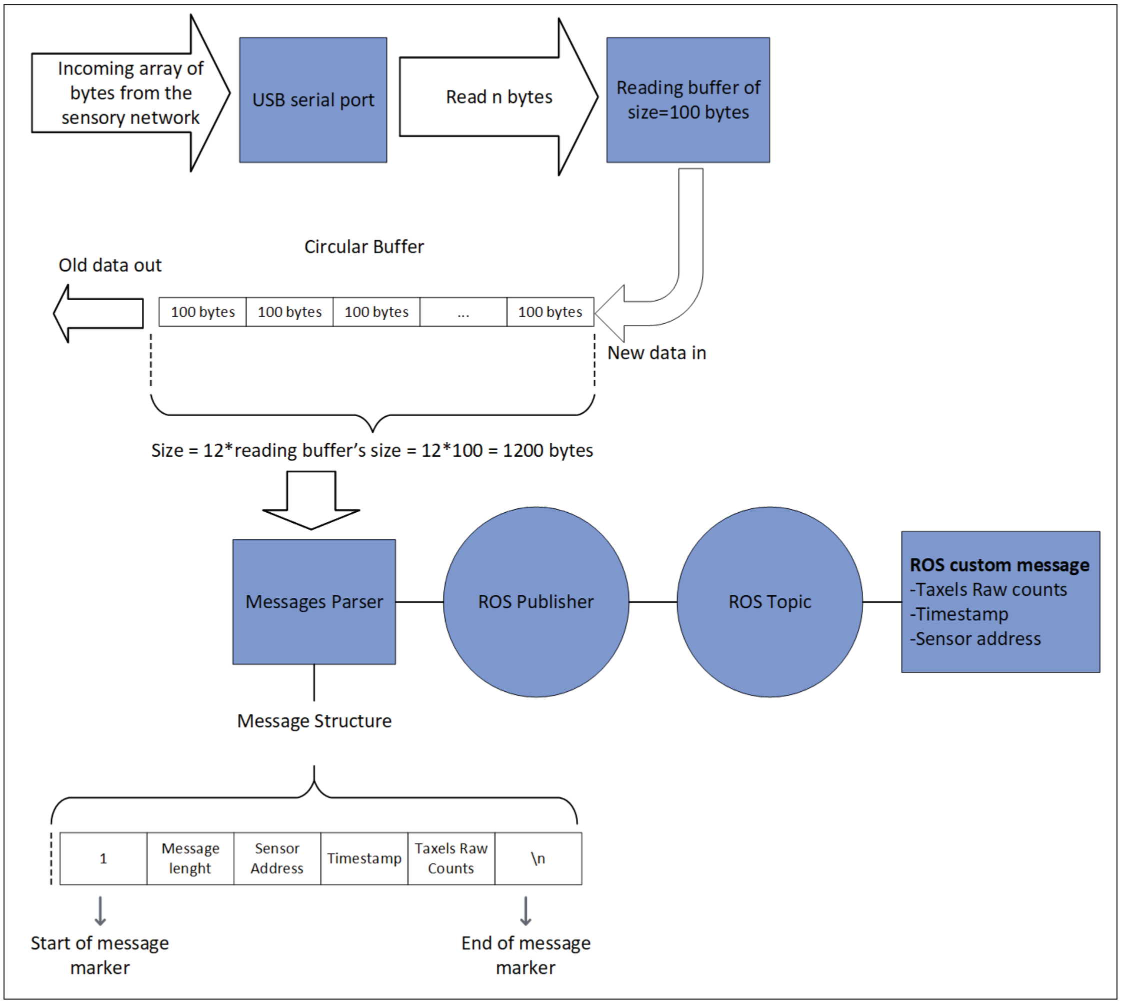


Figure 3.10 I2C Data flow from the sensory network to ROS: The sensory data are cyclically received via USB port. A parser extracts the reported taxels' tactile information from these data and sends them to a ROS publisher

After receiving the data from the sensory network through a USB connection, a c++ program is used to setup a serial port that reads the incoming data into a reading buffer that was set to the maximum allowable size. From there, the data is fed into a circular or ring buffer that can hold multiple incoming messages at a time. Moreover, after reaching its full capacity, this buffer accommodates the new data by cyclically moving its elements in a FIFO or (First-In, First-Out) way to allow for a continuous data flow while maintaining a fixed allocated memory size. In other words, each new batch of read bytes gets pushed into the end of this buffer which forces the oldest data at its start to get discarded. Circular buffers offer many advantages, including an efficient memory usage given their fixed size which discards the need for dynamic memory allocation and therefore prevents memory leaks and fragmentation. Moreover, in our case the circular buffer's ability to retain multiple messages at the same time before parsing them allows to catch more sensory input and prevent data loss. Afterward, the data is parsed into a sensor address to identify the source of the tactile data, a corresponding timestamp and the raw counts for the corresponding taxels. This information is finally published on the ROS topic dedicated to received data's originating sensor.

3.3 CAD Model

After building the sensory data acquisition and parsing system, the Allegro hand along with the sensory network were mounted on a robotic arm. The robotic arm was installed on a workstation that was purchased from Vention and was readily available in the CoRo (*Control and robotics*) laboratories. This setup was prepared in order to place the test objects on the workstation and conduct the objects grasping experiments that will be thoroughly discussed in chapter 5. But before reaching the experiments phase and in order to control the robotics system and acquire all the needed data, an accurate CAD model had to be created for the entire robotic system. The model consists of three major parts: The workstation, the robotic arm and the robotic hand used for objects grasping.

The CAD model for the used workstation was custom created on Vention's official website and the models for the objects to be grasped were created using Onshape and were attached to this workstation.

Among the various robotic arms available in the laboratories, we decided to use the UR5-e robotic arm from UR (*Universal Robots*). The CAD for this arm was retrieved from Universal Robots Official Website (Robots, 2023).

Finally the Allegro hand, a 16 DoF (*Degrees of freedom*) torque controlled robotic hand was employed. The CAD of the hand was supplied upon request from Wonik Robotics company (Robotics, 2023). As previously disscused in this chapter, the capacitive sensors were printed on FPCBs and attachable brackets were designed to mount them on the Allegro hand. The brackets

CAD models were designed in Onshape for each of the 20 different sensors that were used to cover the Allegro hand. On top of these brackets models, the sensors' FPCB electrodes layouts that were designed in Altium were projected and used as blueprints for placing the taxels or tactile sensing units that were represented by small rectangular entities attached to the brackets.

The setup of the complete robot is shown in figure 3.11. Moreover, in figure 3.12 the CAD assembly of the Allegro hand with the installed brackets is shown on the left side and the taxels entities are shown on the right side.

3.4 Robot Control and Sensory Data Handling

In order to control the previously described robotic system, its CAD model has to be converted into a URDF(*Unified Robot Description Format*). URDF is an XML-based file format that is used to describe the kinematic structure of robots and specify their physics and dynamics properties. More specifically, in this type of files a robot is represented as a hierarchical structure of of links that possess both visual appearances and collision geometries and properties like mass, inertia etc. can be attributed to them. These links are connected using joints whose properties like joint type (i.e. revolute, prismatic etc.), joint limits, friction etc. can also be specified. Our robotic model has a total of 22 revolute joints and in order to control them, a URDF description file is essential. The strategy that was followed to control our robotic system in real-life is depicted in figure 3.13.

The control of the entire robotic manipulator can be divided in two major parts: The control of the UR5-e robotic arm whose control box was connected to the controlling computer using an ethernet cable and the control of the Allegro robotic hand that was connected to the computer using a PCAN-USB adapter.

For the first part, since we were interested in controlling the position and the orientation of the end effector of the robotic arm, we decided to control the UR5e using MoveIt. This robotic manipulation and path planning software was used as the top control layer to perform all the kinematics calculation needed to bring the robot's end effector from location A to location B in

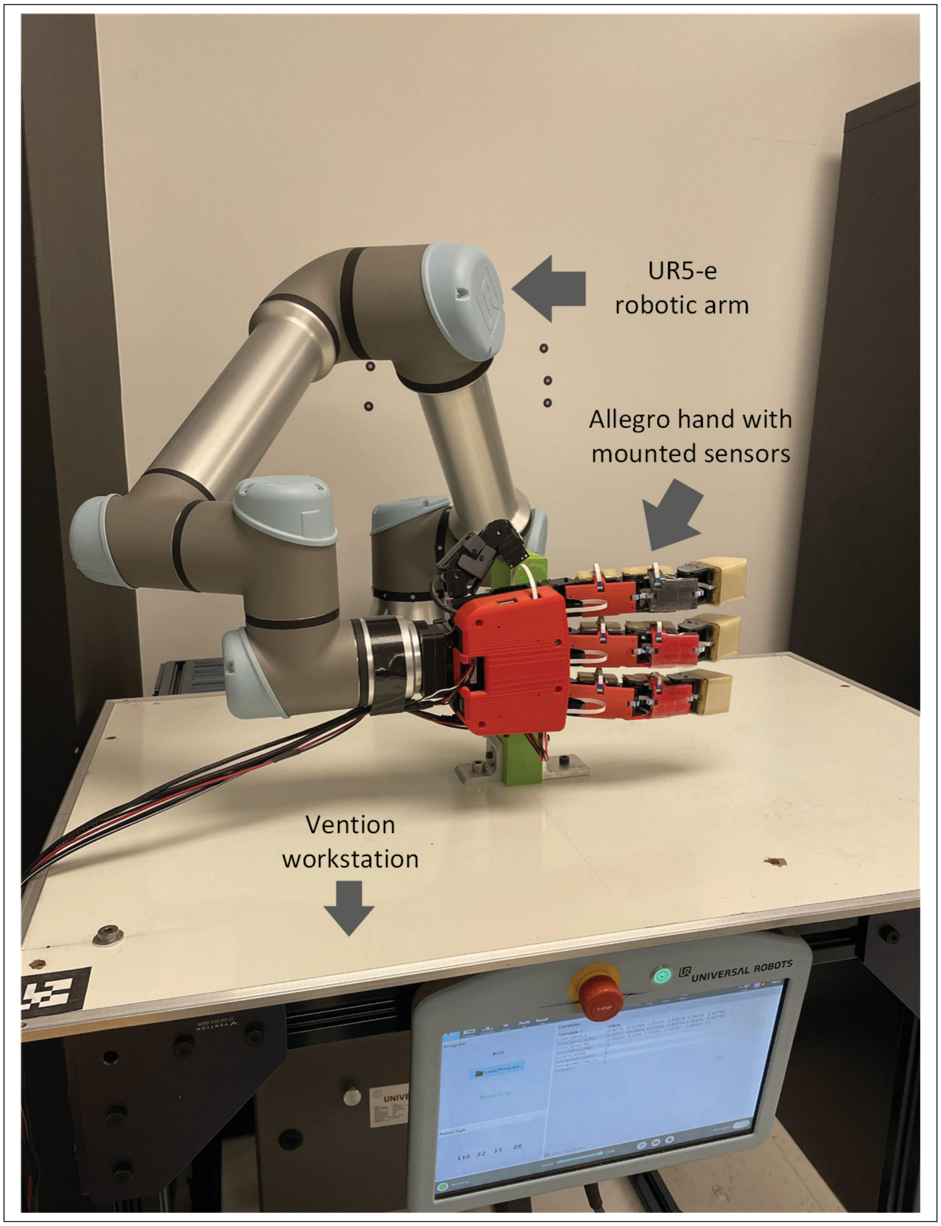


Figure 3.11 Real-life robotic workstation setup with a test object to grasp

the 3D space. Moreover, MoveIt API offers the ability to control the robotic arm using python scripts and this is necessary to automate the grasping experiments that will be discussed in chapter five. However, Moveit cannot directly interact with the UR5e robotic arm and therefore a series of steps was taken to make this interaction possible.

One of the interesting features offered by Universal Robots is URCaps which are plugins

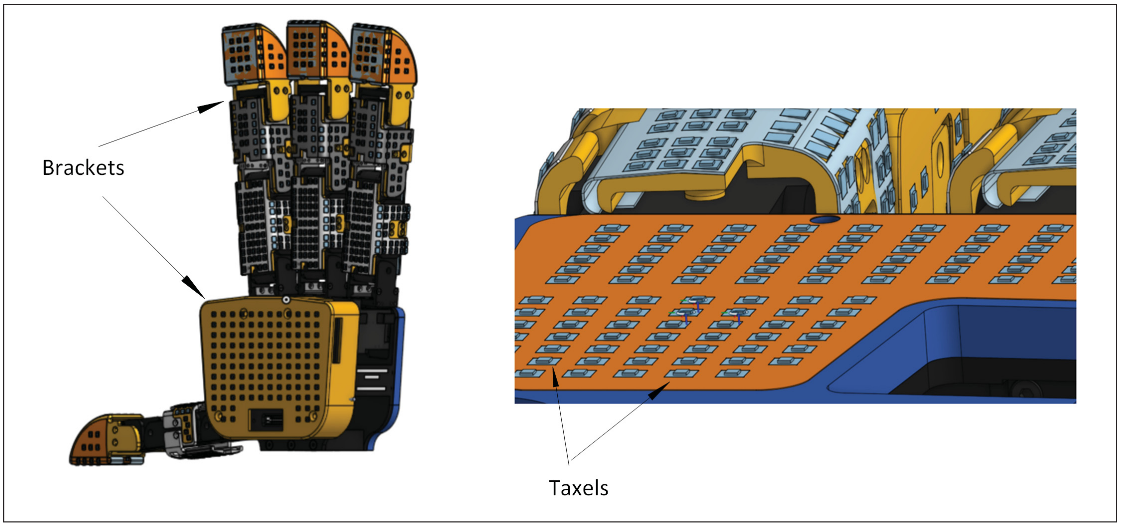


Figure 3.12 Simulated Allegro hand and taxels: The taxels are represented by small rectangular entities to track their 3D coordinates

or software packages that extend the functionalities of the UR robots by allowing them to communicate and interact with external hardware or software components such as sensors as well as programmable logic controllers, software development kits, etc. So first, in our case the external control URCap was used to allow the control and integration of the robotic arm with the external computer. More specifically, this URCap acted as the user interface for the UR ROS2 driver inside the CB3 control box of the UR5e robot.

Second, the ros-control framework was used as the intermediary layer between the URCap and the MoveIt software. This framework offers various types of controllers. In our case, three types of controllers were implemented: First, the joint position controller was used to command the desired positions of the UR5e joints using PID (*Proportional-integral-derivative*)controllers. In the case of our robotic arm, six different revolute joints were commanded by a joint trajectory controller that commands a joint to follow a series of trajectory waypoints within a specific time frame based on the received target position command to provide a smooth motion with reduced jerky movements that were not favorable in our later to be discussed grasping experiments. Second, the joint state controller was responsible for reporting the joint states of the robotic arm at all times and finally the robot status controller was responsible for reporting the robot status at all times and this includes: e-stopped, in motion, accepting motion etc. This is an

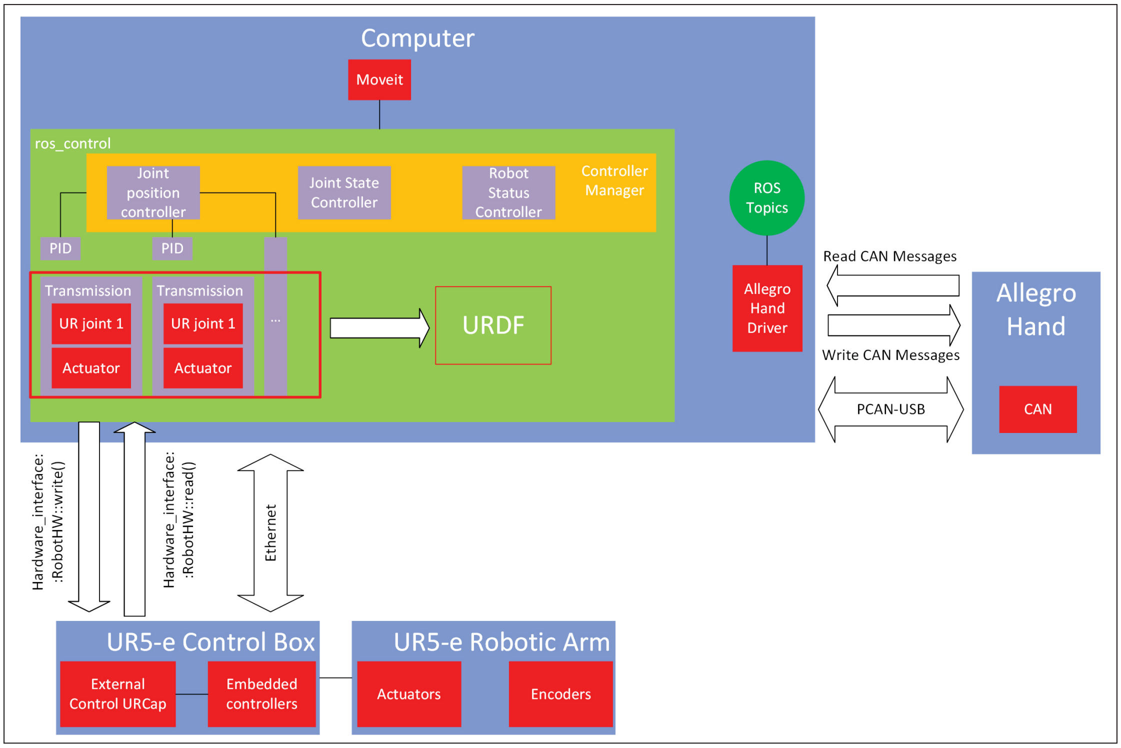


Figure 3.13 Real-life Robot control strategy: the ros-control framework includes controllers and hardware interfaces to communicate with the UR5e robotic arm while exposing the control to the manipulation and path planning software MoveIt. As for the allegro hand, ROS topics are used to directly command the Allegro robotic hand. The controlled robotic system is visualized in RViz

essential piece of information before sending commands through the joints controller. All of the controllers are managed by a controller manager that dictates the life cycles of these controllers and handles resource conflicts among them.

As previously mentioned, the URDF file for the robotic system is crucial for the control scheme since this file defines the kinematics and dynamics description of the entire robot while specifying the the properties of the corresponding joints to be controlled. Moreover, ros-control offers many packages and software libraries to address many hardware related functionalities and some of which require a declaration in the URDF file of the robot. For instance in our project each of the robotic arm joints was coupled with a rotary motor actuator and wrapped within a direct drive transmission element in the used URDF file. Moreover, the joints physics and dynamics properties such as position, velocity and torque limits, internal friction and damping were also declared in the URDF file.

Finally, one of the most important features of the ros-control framework is the provided hardware abstraction layer known as **hardware_interface::RobotHW** class which models the robot's real or simulated hardware resources such as actuators, sensors and encoders. In other words using this class in our project, commands like desired joint positions or efforts can be sent to the previously mentioned external control URCap which communicates them with the corresponding embedded controllers in the CB3 control box of the UR5e arm in order to finally move the corresponding arm's actuators. Also, information like joint states can be received from the robot's encoders and communicated through messages provided by this hardware abstraction layer. This data traffic is indispensable for the implemented feedback-based PID controllers.

As for the second part of our implemented control scheme, the previously mentioned PCAN-USB adapter was used to allow the controlling computer to directly communicate with the CAN (*Controller area network*) that connects all the electronic control units inside the Allegro hand. An allegro hand driver was used to handle the CAN messages communicated with the Allegro hand and transfer their data through ROS topics. More specifically, the Allegro joint positions received from the hand's encoders were published to an Allegro joint states topic and 16 separate topics were used to command the Allegro's 16 joints through position and torque PID controllers in order to generate the adequate PWM signals to the hand's electric motors and generate the desired behaviour. It is worth mentioning that PID position controllers were used to control the fingers bases and maintain their positions throughout any specific grasp, while torque controllers were used to control the rest of the joints since they were needed to fully open and close the allegro hand.

During the robot's control and operation, the sensory readings were monitored at all times. The activation of each taxel was decided based on its raw counts data variation. In other words whenever the raw count variation of a certain taxel exceeds a threshold value of 150 counts, it is considered as active otherwise it is considered non-active and not experiencing any significant contact. This value was selected to be slightly above the sensory signals variation levels that were introduced by noise during the absence of contact. This way, we avoid missing a generated contact and confusing a genuine contact for noise. Moreover, it was essential in our project to track the positions of the taxels in the 3D space in order to generate semantic tactile data. Towards this goal the previously discussed CAD model of the entire robotic system was used to approximate the 3D coordinates of the centers of the sensory taxels with respect to the brackets frames. Given that the FPCBs' taxels are rigidly fixed to the plastic brackets, the aforementioned coordinates remain constant regardless of the robotic system's motion and as such these values were registered in a data structure. Moreover, one of the interesting packages offered by ROS is tf2. tf2 is a powerful library used for managing coordinate transformations in a robotic system and it was used in our project to broadcast coordinate transformations between all the connected parts of our robotic workstation, from the table to taxels on the Allegro hand surface. Moreover this library can automatically perform the proper mathematical calculation to generate the needed transformation between any two parts that are connected in a kinematic chain or tree. Some parts of the tf2 kinematic chain that was generated for our robotic system is depicted in figure 3.14.

As we can see, the different workstation components from the workstation table to the different links of the robotic arm and the finger parts of the Allegro hand are inter-connected in a hierarchical structure and a complete kinematic chain from the uppermost world frame down to a sensor's bracket is highlighted. Between each two connected parts, a coordinates transformation is being constantly broadcast by the robot-state-publisher ROS node.

Therefore, upon registration all the mathematical calculations needed to express the coordinates of the brackets located at the bottom of the kinematic tree with respect to the world frame located at the top of the tree, are performed. And in order to keep track of the positions of the centers of all the taxels that are mounted on the allegro hand with respect to one common static frame of reference, which is the world frame attached to the UR5-e workstation's table in this case, the already calculated coordinates transformations between the world frame and the brackets are applied on the already registered taxels coordinates with respect to their corresponding brackets.

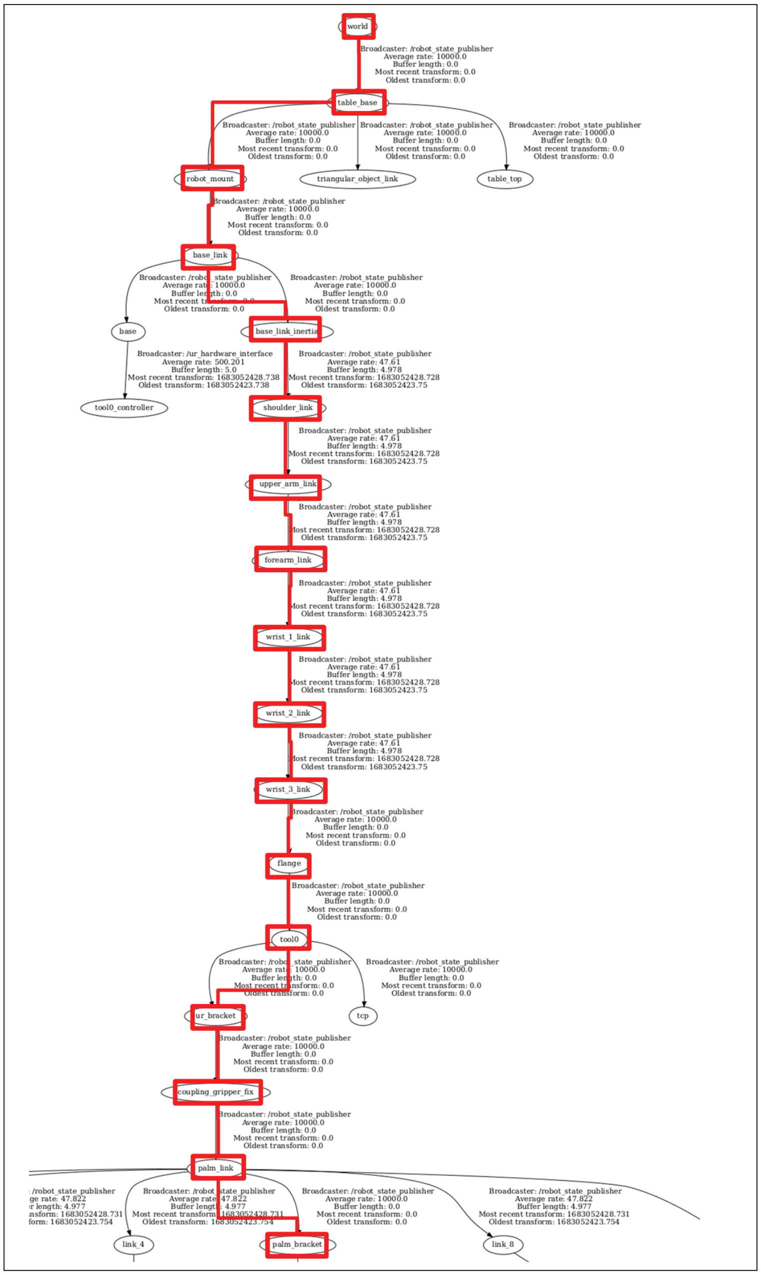

world
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
table_base
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
robot_mount
triangular_object_link
table_top
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
base_link
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
base
base_link_inertia
Broadcaster: /ur_hardware_interface
Average rate: 500.201
Buffer length: 5.0
Most recent transform: 1683052428.738
Oldest transform: 1683052423.738
Broadcaster: /robot_state_publisher
Average rate: 47.61
Buffer length: 4.978
Most recent transform: 1683052428.728
Oldest transform: 1683052423.75
tool0_controller
shoulder_link
Broadcaster: /robot_state_publisher
Average rate: 47.61
Buffer length: 4.978
Most recent transform: 1683052428.728
Oldest transform: 1683052423.75
upper_arm_link
Broadcaster: /robot_state_publisher
Average rate: 47.61
Buffer length: 4.978
Most recent transform: 1683052428.728
Oldest transform: 1683052423.75
forearm_link
Broadcaster: /robot_state_publisher
Average rate: 47.61
Buffer length: 4.978
Most recent transform: 1683052428.728
Oldest transform: 1683052423.75
wrist_1_link
Broadcaster: /robot_state_publisher
Average rate: 47.61
Buffer length: 4.978
Most recent transform: 1683052428.728
Oldest transform: 1683052423.75
wrist_2_link
Broadcaster: /robot_state_publisher
Average rate: 47.61
Buffer length: 4.978
Most recent transform: 1683052428.728
Oldest transform: 1683052423.75
wrist_3_link
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
flange
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
tool0
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
ur_bracket
tcp
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
coupling_gripper_fix
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
palm_link
: /robot_state_publisher
ge rate: 47.822
r length: 4.977
nsform: 1683052428.731
form: 1683052423.754
Broadcaster: /robot_state_publisher
Average rate: 47.822
Buffer length: 4.977
Most recent transform: 1683052428.731
Oldest transform: 1683052423.754
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
Broadcaster: /robot_state_publisher
Average rate: 47.822
Buffer length: 4.977
Most recent transform: 1683052428.731
Oldest transform: 1683052423.754
link_4
palm_bracket
link_8

Figure 3.14 Portions of the tf2 tree showing the hierarchical connection among the different parts of the robotic workstation that was used during our grasping experiments

Therefore at any given moment, the 3D positions of the centers of the active and non-active taxels with respect to the static world frame can be registered and a corresponding point cloud can be generated.

CHAPTER 4

SENSOR SIMULATION

4.1 Introduction

As previously mentioned in the introduction, the desired outcome of the simulation is to create a robotic model equipped with tactile sensing abilities that can be controlled to replicate the developed capacitive sensor's behaviour and produce reliable synthetic tactile data that could be used later on for training artificial intelligence models for objects recognition. Towards this objective, a digital representation for the robotic workstation that was previously described in chapter three was created and and placed in the open source Gazebo simulation environment to simulate the physics of the robot's operation during the grasping experiments that will be discussed in chapter five. In this chapter, we present the simulation setup for our specific application , we describe the followed methodologies to control the simulated robot and to generate the synthetic tactile data during grasping and finally we go over the simulation's imperfections while giving potential fixes.

It is worth mentioning that a literature review has been already conducted and included in chapter one to compare Gazebo with other simulators for robotics applications.

4.2 Simulation Setup

4.2.1 Modified CAD and SDF (Simulation description format) Models for the Robotic System

In order to simulate the robotic system that was previously described in chapter three, a proper digital representation must be created and placed in the simulation environment.Towards this goal, first the CAD model that was previously used in chapter three has to be modified given that it was previously created for the sole purpose of controlling the real-life robotic setup. However since the main purpose of the simulation is to create reliable synthetic tactile data that are as close as possible to the real-life data, a digital representation of the sensor setup that was previously discussed in chapter three should be built. While the taxels were represented by rectangular entities that allowed tracking their 3D positions at all times, the dielectric layer that covers the sensor was absent in the model. Ideally speaking, a layer that reflects the dielectric's elastic behaviour should be added on top of the taxels. However, since it is a rigid body simulator Gazebo is not equipped with the tools to simulate the aforementioned behaviour. Therefore a rigid layer with a specific thickness seemed enough for our project. The thickness of the simulated layer was approximated by that of the real-life dielectric layer while undergoing an external force of 50 N and in this case it was around one mm. We acknowledge that the external forces applied to the sensor during the grasping experiments that will be discussed in the chapter five are not 50 N for all contacts at all times and other thickness values could have been adopted, however for our application the current selection was considered a good approximation. However, Gazebo could not accurately detect the contact of the simulated dielectric layer with external objects when the contact covered relatively large surfaces because of the big amount of contact points that needs to be processed in this case. Therefore, in order to reduce the amount of generated contact points and allow Gazebo to process them more accurately, the dielectric layer was instead represented by a pattern of micro-structural cones placed on top of the taxels. More specifically the center of each cone was perfectly aligned with the center of its corresponding taxel. The microstructural cones placed on top of the Allegro hand's palm in the simulation environment are shown in figure 4.1.

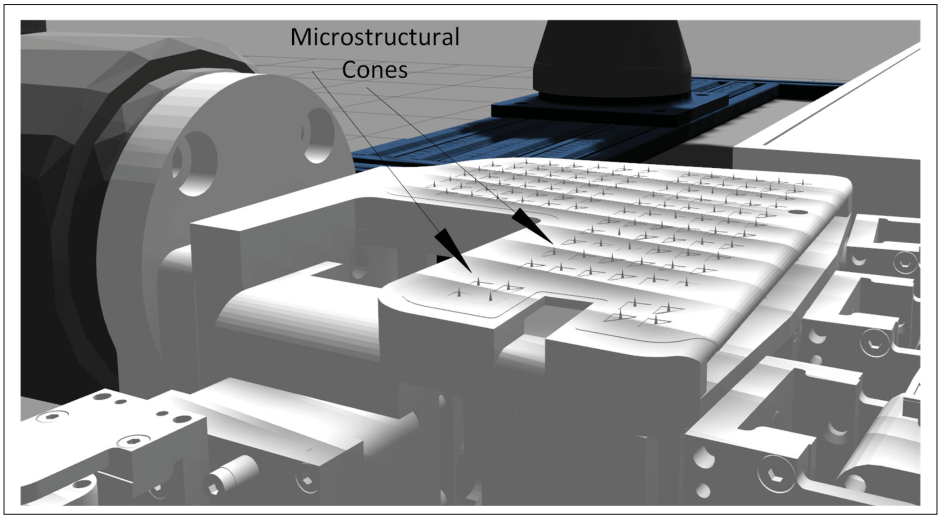


Figure 4.1 The conical microstructural pattern created to simulate the real-life dielectric

Also, the contact points detected by Gazebo when the Allegro hand's flat palm enters in contact with a flat object's face before and after placing the microstructural cones are illustrated in figure 4.2.

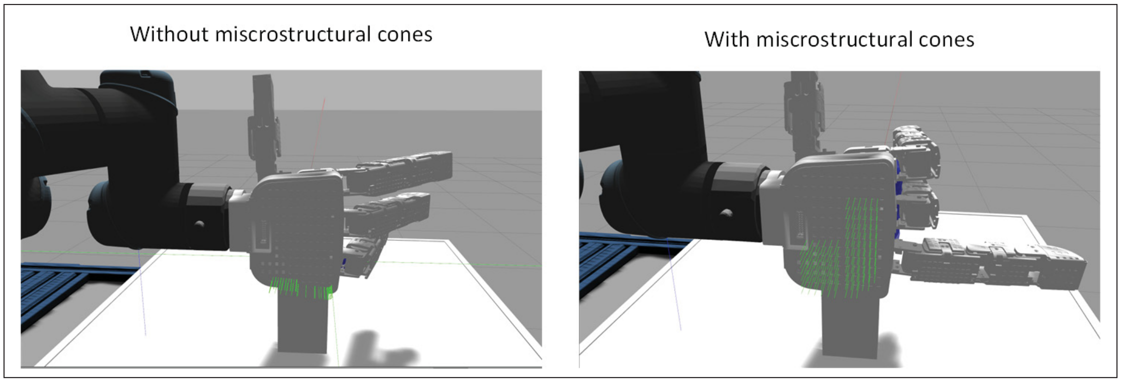


Figure 4.2 Gazebo contact detection over relatively wide collision surfaces: Before adding the microstructural cones, only a non-representative portion of the actual contact between the flat Allegro hand's palm and an external object's flat face is detected. After adding the microstructural cones, the detected contact is more accurate and representative. The contact normals in green reflect the contact on the palm but it is seen from the back of the hand

As we can see, before placing the microstructural pattern on the palm's surface, Gazebo only detects a small portion of the total contact points near the lower borders of the palm. But after placing the microstructural cones, the contact area decreases so Gazebo is able to handle it and report all the points of contact between the cones tips and the external object's flat face leading to a better contact representation. Finally, while Gazebo contact detection limitation is illustrated using the hand's palm the same applies for the rest of the hand's parts.

Afterward, a modified URDF file was extracted from the CAD model and was used once again to control the robot.
Unfortunately, Gazebo supports SDF formats instead of URDF formats. However, the simulator supports a built-in converter that was used to generate a SDF file for our robotic system. Similarly to URDF, the SDF files are XML-based texts that define the robot's visual as well as physics and dynamics properties. However, they also define the world environment and most importantly they allow to define the sensors to add to the model and the plugins to load during the simulation. Plugins are software components that allows to interact with the simulation environment and this includes processing the data generated from the running simulation and add custom behaviours to the simulation environment and the entities that are present inside of it. The plugins and sensors that were added to our robotic model in Gazebo will be thoroughly discussed in the next section.
The simulated setup of the complete robot is shown in figure 4.3 and the workflow using the different file formats is depicted in figure 4.4.

As we can see, the CAD model for the entire robotic system is converted into both URDF and SDF files. As previously discussed, the URDF is essential to declare the controllable joints along with their properties which are then used by the ros-control framework to command the real-life or simulated robotic system.
As for the SDF file, it is crucial to digitally represent the robot in the Gazebo simulation environment. While the SDF files are very similar to the URDF files, they offer additional functionalities. For instance, in our project, readily available simulated tactile sensors were attached to the simulated model's SDF to generate synthetic tactile data during the grasping

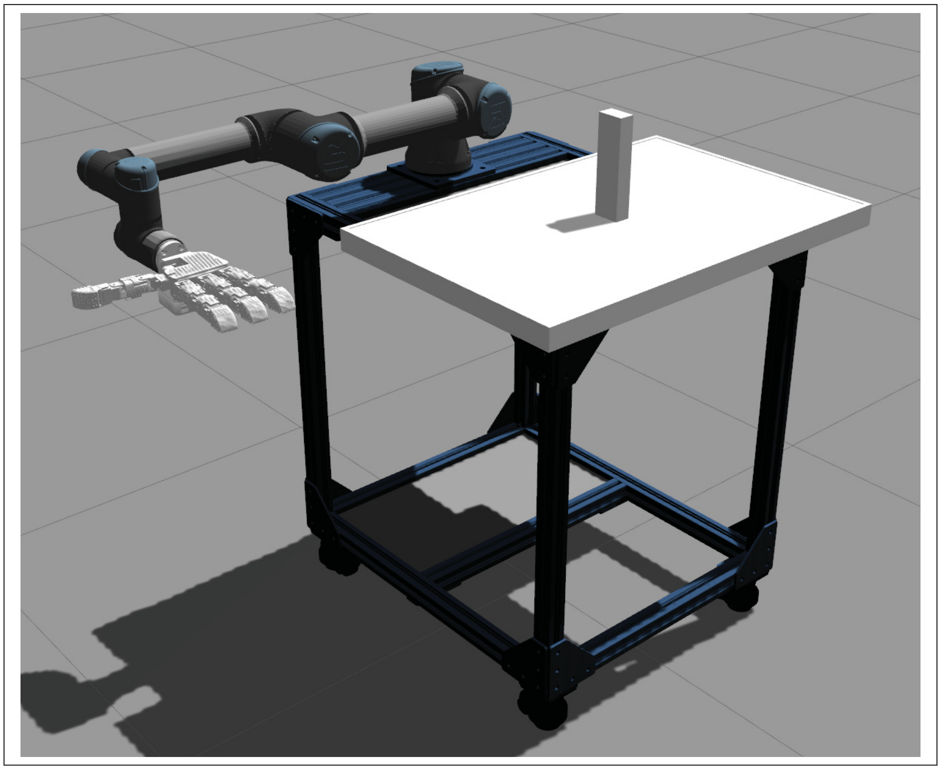


Figure 4.3 Simulated robotic grasping workstation with a test object to grasp

experiments to be discussed later. Also, gazebo-ros-control was attached to the robot's SDF and this is a ROS package that acts as the middle-man between the ros-control framework and the Gazebo simulator.

4.2.2 Robot Control and Sensory Data Handling

The control scheme that was followed for the simulated robot is depicted in figure 4.5.

The simulated robotic system was controlled in the same way the real-life robot was controlled as previously described in chapter 3. However, in this case the Allegro hand was mounted the UR5e robotic arm in the URDF file and its joints were coupled with rotary motor actuators, declared inside direct drive transmissions and their physics and dynamics properties were specified. Moreover, instead of sending the controllers commands to real-life robotic components, the

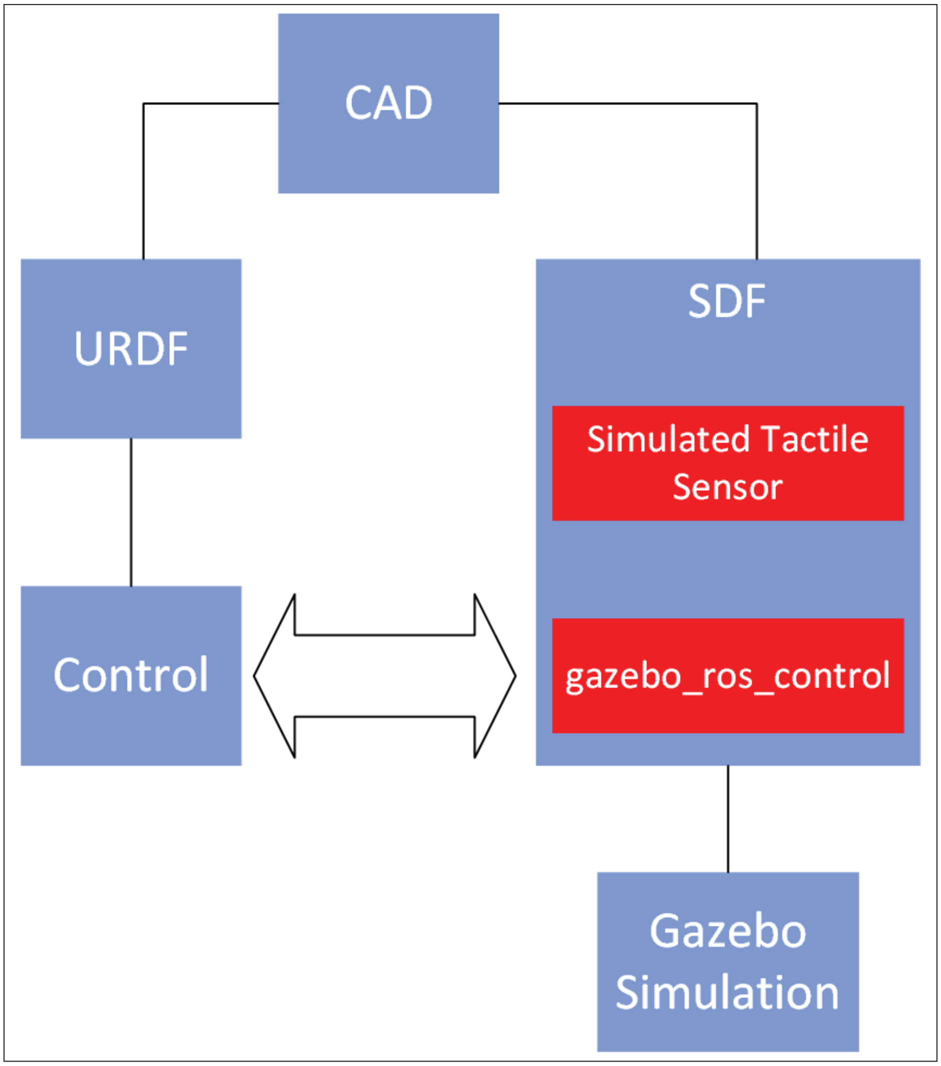


Figure 4.4 The different types of files used in our project and the purpose for each: URDF file was used for control and SDF file was used for Gazebo simulation and both were connected through a gazebo-ros-control plugin

commands were directed towards the simulated robotic system that was represented by a SDF file. The gazebo-ros-control package discussed earlier was attached as a plugin to the robot's SDF model and used to communicate the ros-control messages to the simulated robot.

In order to generate the synthetic tactile data using the Allegro hand, we relied on the collision detection ability that is built-in within the ODE adopted in Gazebo. But no semantic data can be generated solely by ODE's contact information and therefore a filtering algorithm was

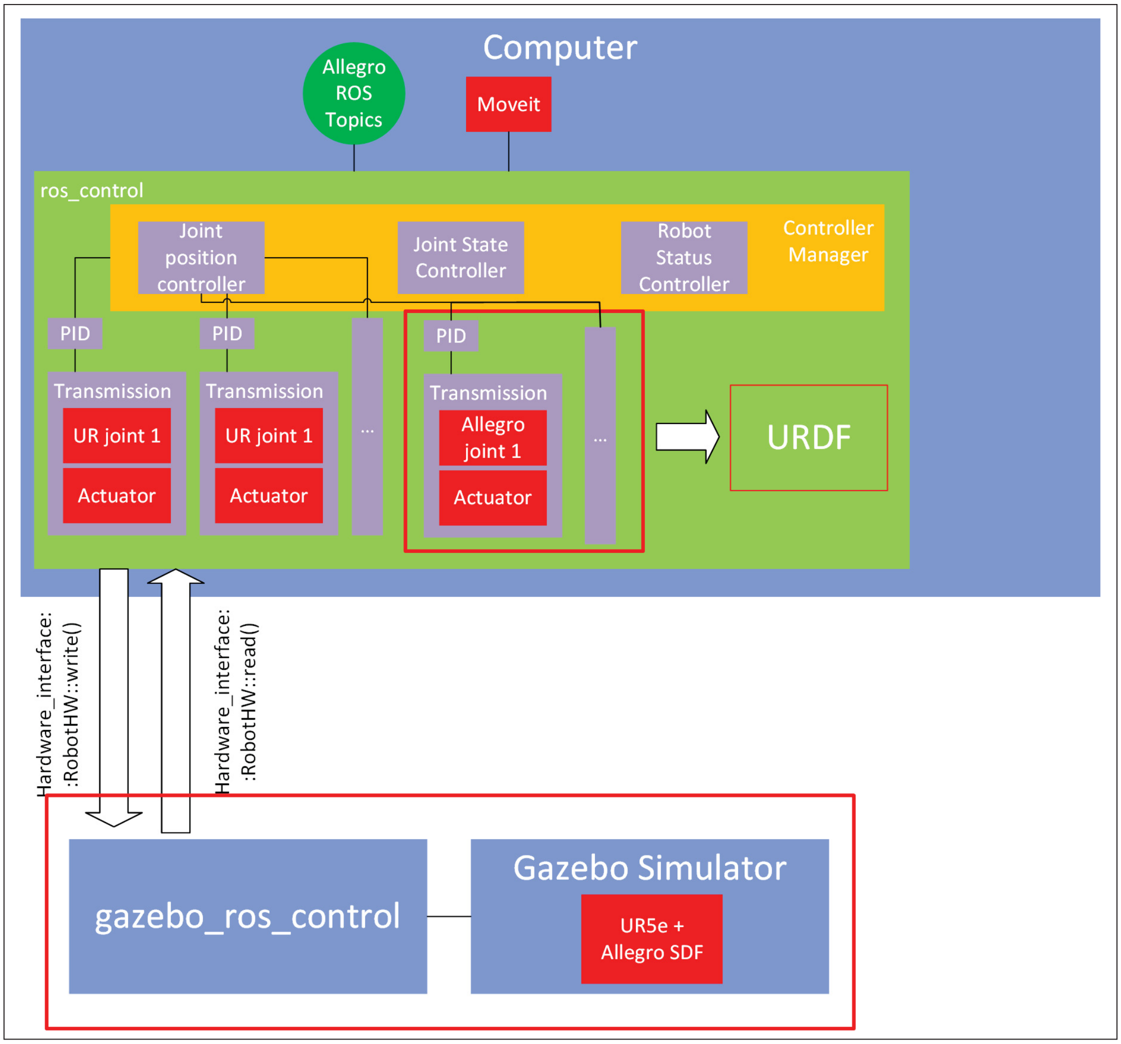


Figure 4.5 Simulation control strategy: the same strategy as the one followed in chapter three to control the robot in real-life was applied in simulation. However, the allegro hand was added to the robotic system's URDF and its joints are commanded using ROS topics that go through the ros-control framework. Also, instead of interacting with the real robotic arm and hand, ros-control interacts with gazebo-ros-control package that is attached to the SDF model of the robotic system in the simulation environment

developed in order to detect the activated sensory taxels on the robotic hand. Figure 4.6 presents an overview of the contact detection pipeline that was implemented in our project.

Among the capabilities offered by Gazebo which are important for this pipeline are the sensor elements and the plugins. In our case, a contact sensor was used and this is essentially an entity

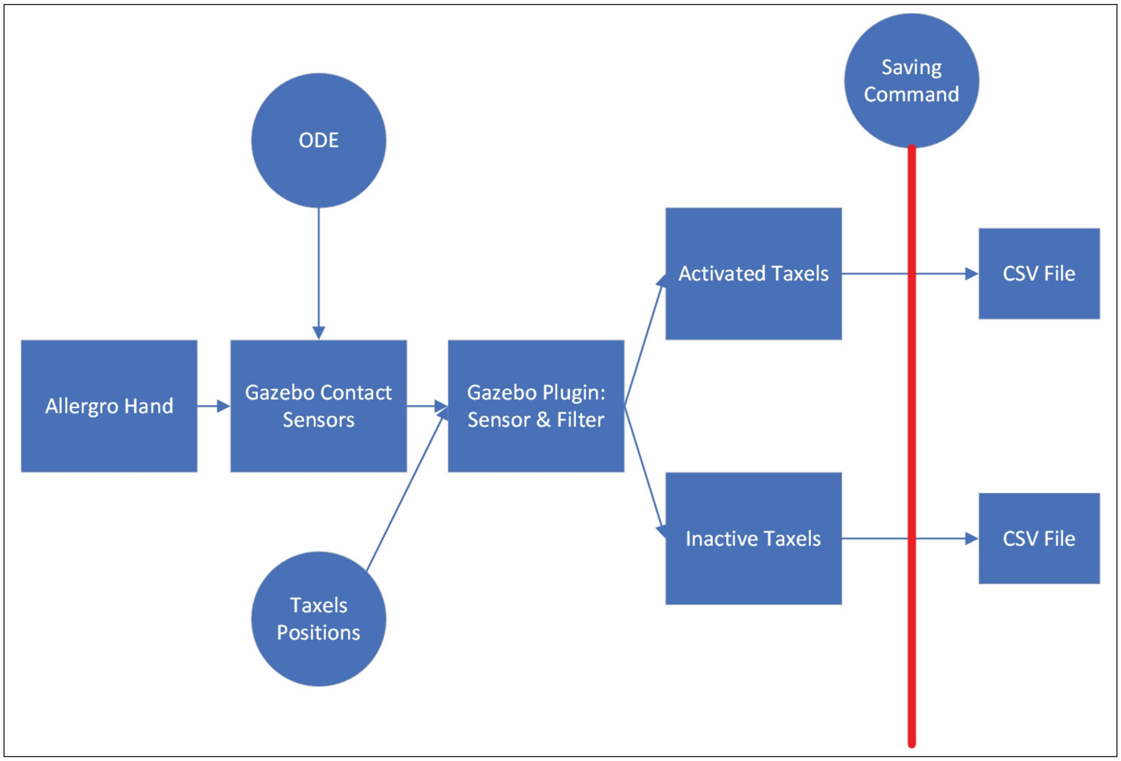


Figure 4.6 Simulation contact detection procedure: Gazebo simulator relies on ODE physics engine to detect the contact points on the Allegro hand upon collision with external objects. These points are compared against the tracked taxels locations to decide which taxels to activate before saving the data into CSV (*Comma separated value*) files

that can be attached to any body in Gazebo in order to publish its contact data to a topic during the interaction with the surrounding simulation environment. Such a sensor is based on the selected the collision detector of the selected physics engine which is ODE in our project. Moreover, Gazebo abilities can be extended using the previously discussed plugins, which are external chunks of C++ code that allow to interact with the simulation environment and control almost any aspect of Gazebo at runtime and this includes: actuators control, sensory data processing, models insertion/deletion etc. In this pipeline, a plugin is used to compare the previously discussed tracked taxels positions to the contact positions reported by Gazebo sensors and finally report the active taxels as 3D coordinates. More specifically, the plugin goes over each of the taxels in each sensor and measures the Euclidean distance between the previously registered taxel's 3D position and the contact position reported by Gazebo. If this distance is below a specified threshold value, the taxel will be registered as active, otherwise the taxel will be registered as inactive. The threshold distance value was calculated as depicted in figure 4.7. Ideally, the distance between the center of the taxel that is supposed to be activated and the contact point reported by Gazebo is supposed to be equal to one millimiter and while any other value that is slighly higher than one millimiter would have worked well, the used threshold distance was calculated in a way to cover a cone with a base radius that is equal to half of the diagonal distance of a single taxel. The higher threshold distance value previously calculated in figure 4.7 not only provides a safety margin in reporting the active taxels without compromising the accuracy of the data but its calculation method also provides us with the flexibility to emulate the contact propagation effect in the real-life sensor's dielectric by controlling this threshold distance value. In other words, in the same way followed in figure 4.7 we can calculate a threshold distance value for a cone whose base goes beyond the corresponding taxels and covers multiple neighbouring taxels and in this case not only we activate the taxel that is directly located under the microstructural cone but we also activate its immediate surrounding.

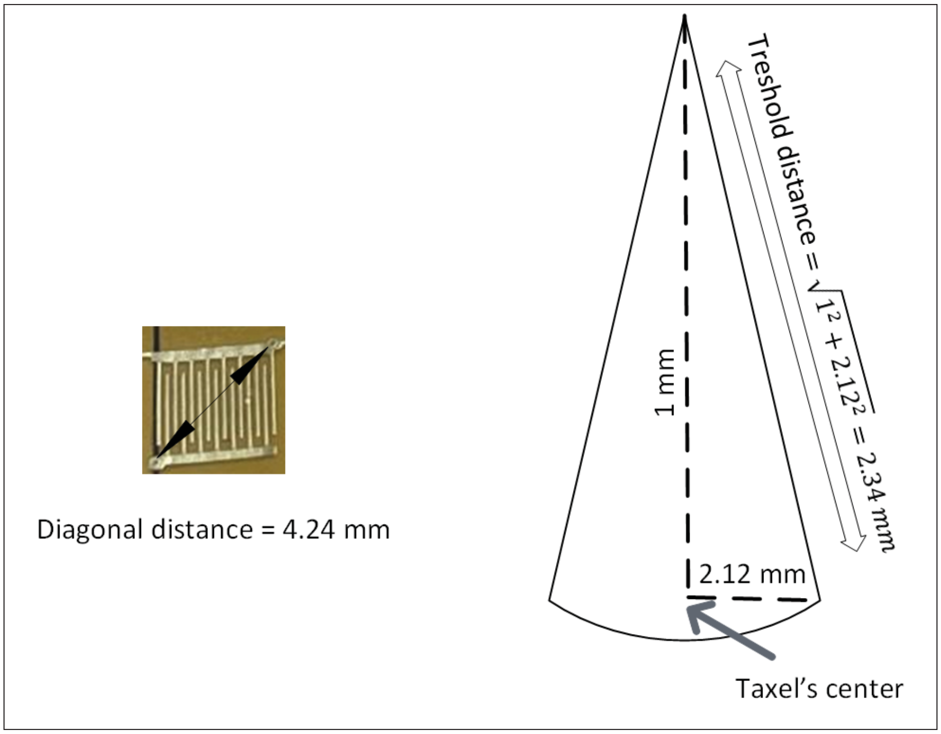

Treshold distance = $\sqrt{1^2 + 2.12^2} = 2.34$ mm
1 mm
2.12 mm
Taxel's center
Diagonal distance = 4.24 mm

Figure 4.7 Simulation contact filter distance threshold: In simulation, the dielectric is represented by a conic mistrostructural pattern and the distance threshold is calculated as the distance between the simulation contact point, which is ideally located at the tip of the cone, and any point on the circumference of the cone's base

4.3 Simulation Imperfections

Initially, many factors in the created simulation model have increased the gap between the real and synthetic generated data and while some have been accounted for, others are yet to be solved in the future. For instance, the dielectric layer of the developed tactile sensor was not replicated in the simulation along with its elastic properties but rather it was replaced by micro-structural cones on top of the Allegro hand brackets to account for Gazebo limitations when it comes to contact points in case of collisions that cover relatively large surface areas. But in the future, a model can be created to properly simulate the dielectric using another simulator.

Finally, as it will be discussed in chapter 5 despite the meticulous building process, the real-life setup of the workstation cannot be completely reflected in the simulation in terms of positioning and measurements, for example the six dimensional pose of the object to be grasped during the experiments cannot be perfectly reflected in simulation. Therefore, a proper calibration process that will be discussed in chapter five was performed to rectify this issue and reduce the simulation to reality gap.

CHAPTER 5

EXPERIMENTAL SETUP AND RESULTS

5.1 Introduction

As outlined in the introductory section, this thesis aims to achieve two primary objectives: the development of tangible sensors in both physical and simulated forms. This undertaking necessitates the generation of authentic real-world and synthetic data, enabling a comprehensive assessment of the efficacy of each approach and facilitating a comparative analysis between them. In this chapter, we first describe the data generated in both real-life and simulation then we go over the experimental configurations employed to generate each and we present our data analysis along with the qualitative and quantitative comparison before finally discussing the applied data treatment techniques that allowed us to bridge the sim-to-real gap which is essential to create both semantic and realistic synthetic tactile data.

5.2 Data Description

For the performed experiments, a total of three different test objects of primitive shapes were used: a rectangular prism, a cylinder and a triangular prism. All the 3D printed test objects are depicted in figure 5.1. Each object was grasped from different AoAs (*Angles of approach*) including from the top and the sides, to insure a comprehensive tactile interaction with all the object's surfaces. Moreover, each grasp from a specific angle was performed multiple times. Table 5.1 presents all the performed grasps. It is worth mentioning that for the performed experiments we were only interested in the taxels installed only on the frontal side of the allegro hand, therefore the tactile sensors on the backside were totally ignored. However, it is worth noting that these backside sensors have been employed in the previously discussed work by Thomasson *et al.* (2022) to classify object motions in cluttered environments. Moreover because of connectivity issues the middle and pinkie fingertips sensors were providing unstable readings and were therefore ignored as well. These connectivity issues are further discussed along with suggested solutions in the conclusion.

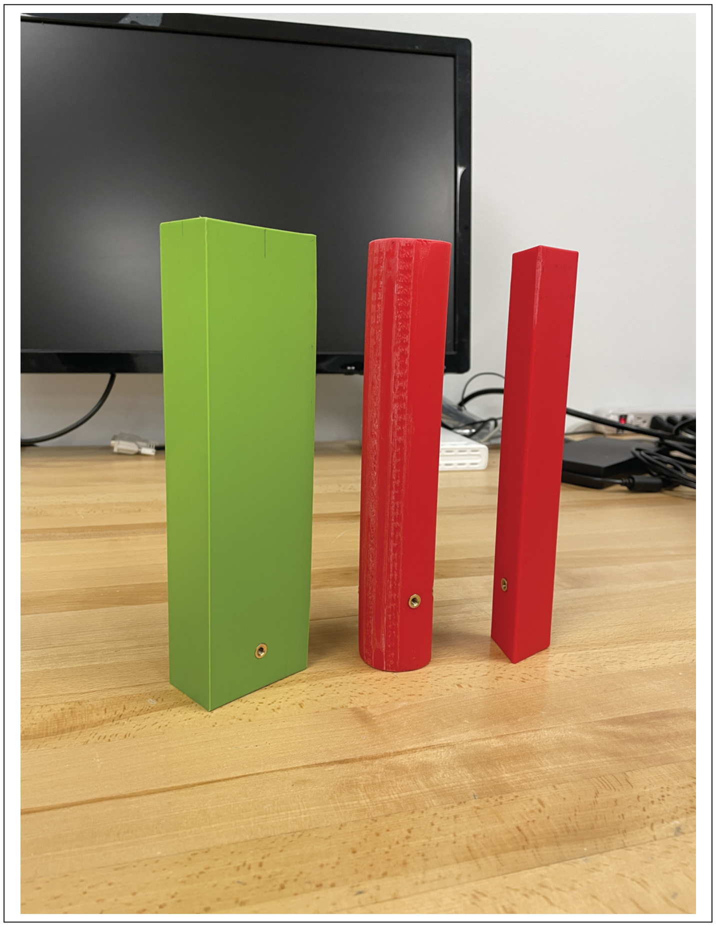


Figure 5.1 The three test objects used in the grasping experiments

Table 5.1 Experiments grasps

Object shape	**Grasp AoA**	**Iterations**
Rectangular prism	Wide side face 1	6
	Narrow side face 1	10
	Wide side face 2	8
	Narrow side face 2	10
	Top face	10
	Side edge	10
Cylinder	Side 1	10
	Side 2	10
	Side 3	10
	Side 4	10
	Top face	10
Triangular prism	Side face 1	10
	Side face 2	10
	Side face 3	10
	Top face	10

5.3 Simulation Setup

After spawning the UR5-e workstation in the simulation environment, which was previously described in chapter three, and setting it to the initial state as shown in figure 5.2, Moveit is used to send the robotic arm's end effector to a goal pose that is convenient to grasp the object from a specific AoA. Subsequently, rostopic commands will be sent to properly position the allegro hand's fingerbase joints and close the robotic hand against the test object.

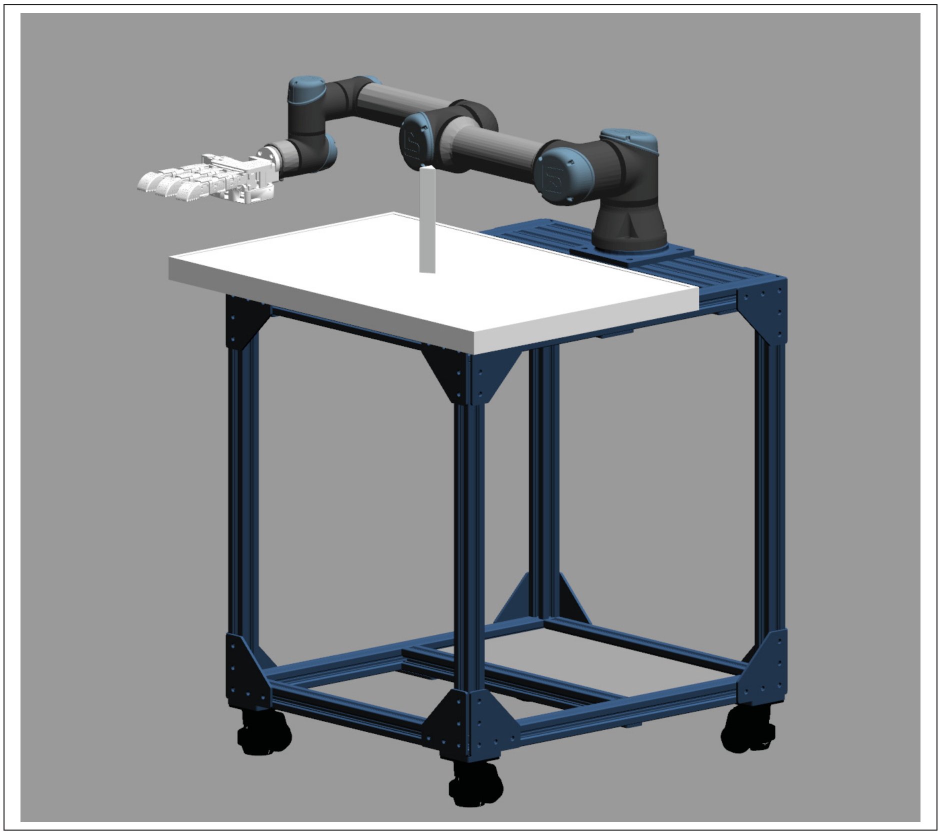


Figure 5.2 UR5e Workstation Initial Pose

Upon hand closure, all the contact point detected by ODE on the allegro hand are registered and compared against the taxels locations to recognize which taxels to activate before finally recording the corresponding tactile point cloud. The point cloud of a grasp against the cylindrical object in both real-life and simulation is depicted in figure 5.3 where both the active and non-active taxels are shown. For each of the grasps previously shown in table 5.1 the workstation is respawned and the same action sequence is repeated. This whole process was automated using a python script.

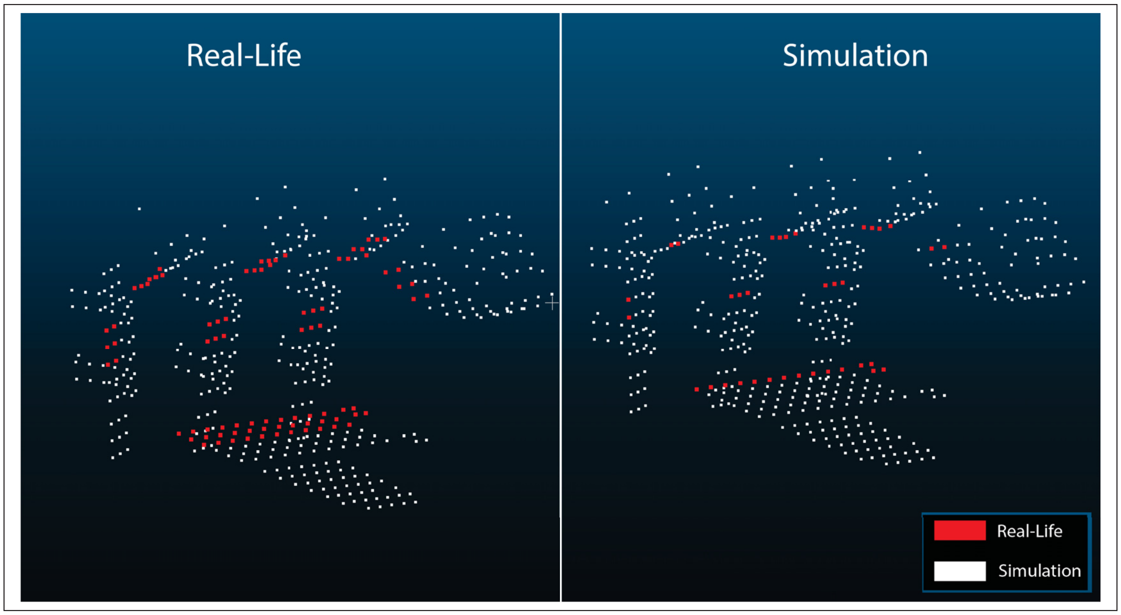
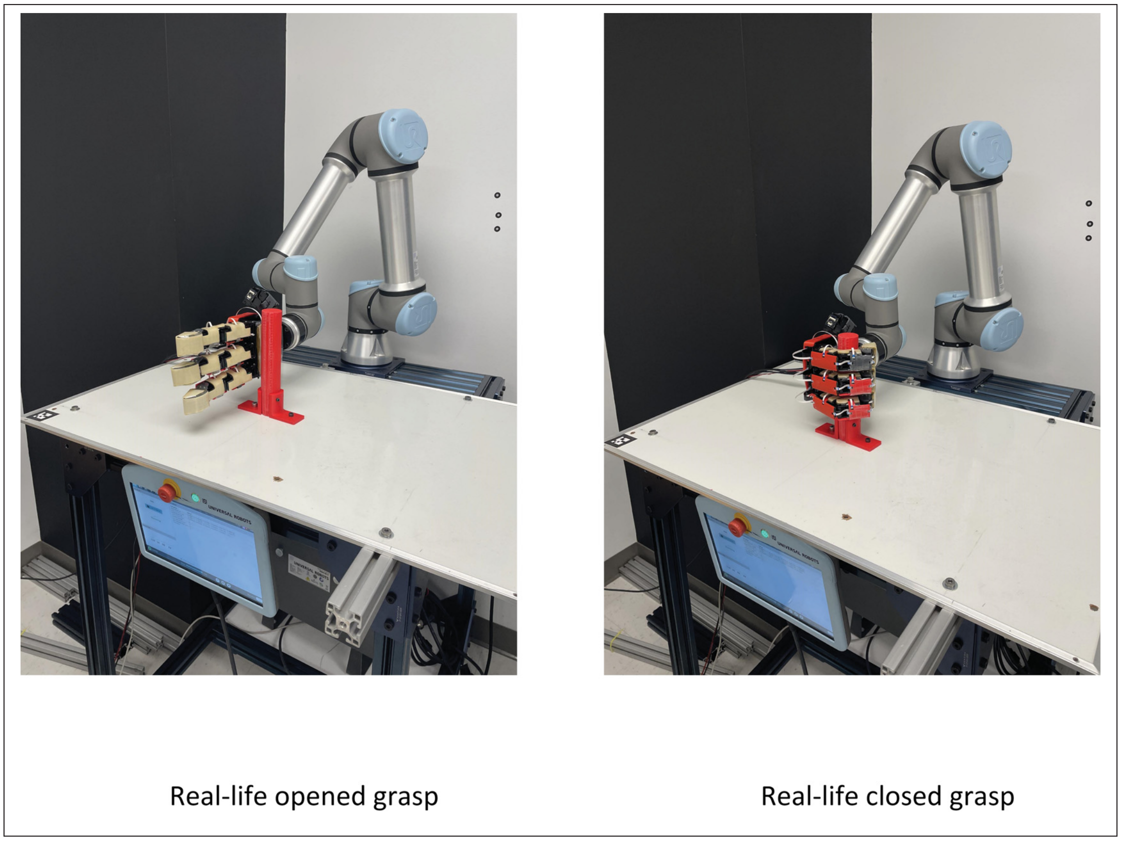


Figure 5.3   Real-life Vs. simulation contact point clouds for the cylindrical prism

5.4 Real-Life Setup

The same data generation procedure followed in simulation was used in the real-life experiments but with minor modifications and can be summarized as follows: After powering up the tactile sensors and setting the workstation to its initial state, an average raw count for each taxel on the allegro hand was calculated based on 100 sampled readings to create a base value before grasping the test objects. Similarly, Moveit is used to command the robotic arm's end effector goal pose for a desired AoA. Afterward, rostopic commands were used to adjust the fingerbase joints of the allegro hand and securely grasp the test object. After grasping the object, this time the taxels activation is decided based on their raw count differential readings as previously discussed in chapter four and the coordinates of both the active and inactive taxels are registered to form a point cloud. Just like in simulation, the process was automated using a python sctipt. One of the real-life grasps is illustrated in figure 5.4



Real-life opened grasp

Real-life closed grasp

Figure 5.4 Real-life Grasp against the cylinder object

5.5 Hardware Calibration

Since the simulation experiments are expected to replicate the real-life scenarios as accurately as possible, a calibration was done to accurately reproduce the details of the real-life setup in the simulation environment, more specifically the 6DoF position of the grasped object relative to the robot's base. This way we make sure that the sources of sim2real errors exclude any setup misrepresentation in the simulation environment and get to focus on the other potential sources of error mainly related to the developed tactile sensor. Towards this end, the C-Track 780 dual camera sensor provided by Creaform was used to measure the coordinates of the test objects with respect to the robotic arm's static base frame in order to accurately replicate the test object's position in the simulation environment. It is worth mentioning that the C-Track's measurement accuracy can go down to the order of micrometers and therefore allowing the real-life grasps to be replicated in simulation with a high degree of accuracy. The C-Track 780 is shown in figure 5.5

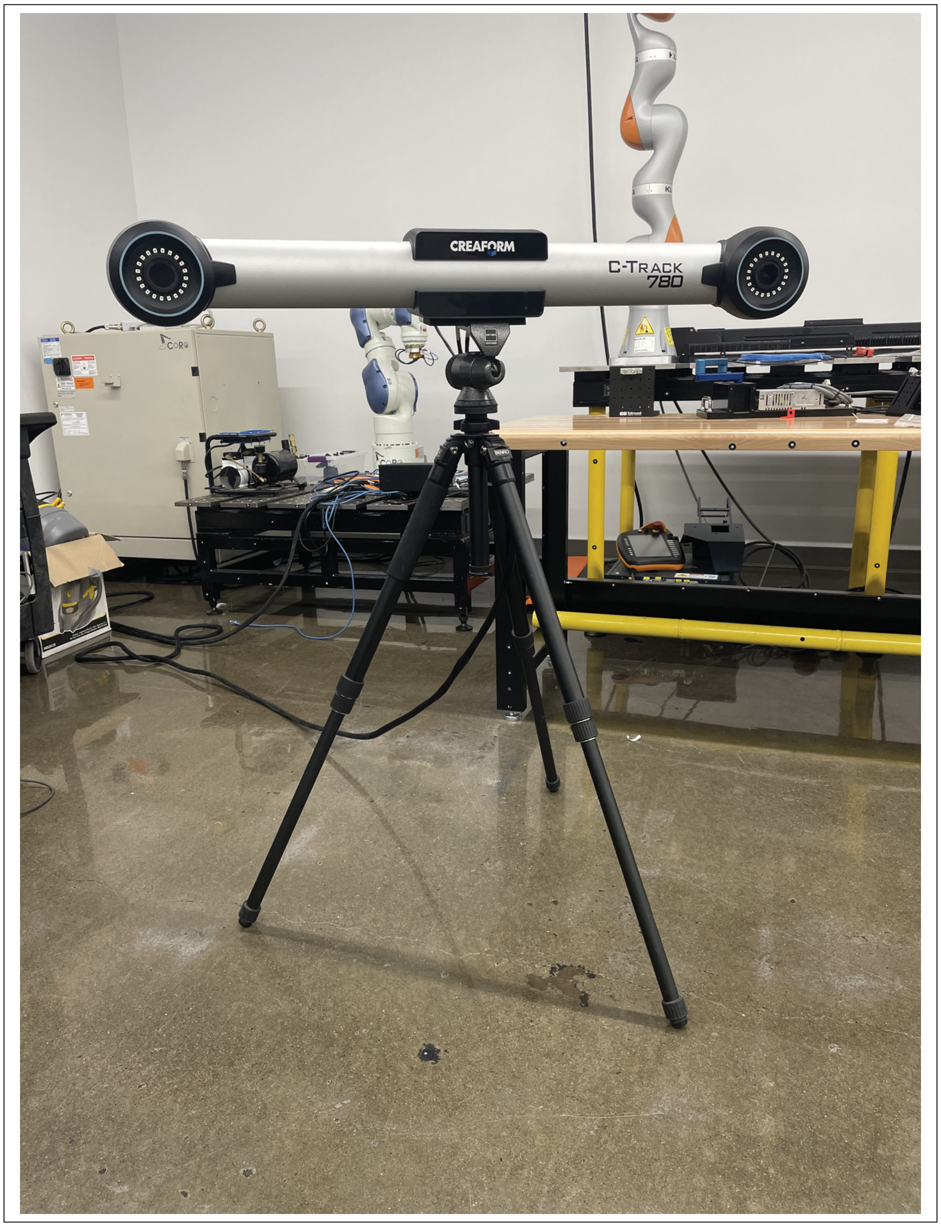


Figure 5.5 C-Track 780

5.6 Results

After generating both the real-life and synthetic tactile data, for each object the contact points collected from all the performed grasps were accumulated and analyzed. A summary of the results is presented in table 5.2. Therefore throughout the whole analysis in this section, the accumulation point clouds were exclusively used.

Table 5.2 Grasping Results

Object shape	Total grasps	Real-life contact points	Simulation contact points
Rectangular prism	54	3504	2092
Cylinder	50	3131	1682
Triangular prism	40	3157	2174

While contact accumulation point clouds were generated for the three different previously mentioned shapes, those of the cylindrical object in both real-life and simulation are depicted in figure 5.6

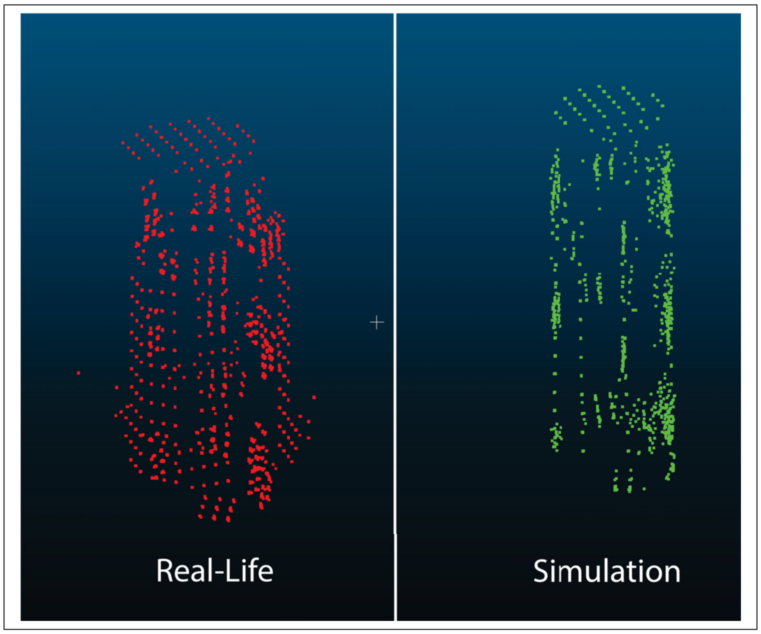


Figure 5.6 Real-life and simulation accumulation point clouds for the cylinder object before data treatment

In order to analyze and compare the real-life to simulation data, a CAD matching was performed on each. In other words, each point cloud was aligned with its corresponding object's CAD model and point to point distance statistics were performed in CloudCompare software. The CAD to point cloud alignment for the rectangular prism is shown for both simulation and real-life data in figure 5.7. Moreover, a matching between the real-life and the corresponding simulation point clouds was performed and the alignment between both clouds for the cylindrical prism before any filtering and adjustment is depicted in figure 5.8

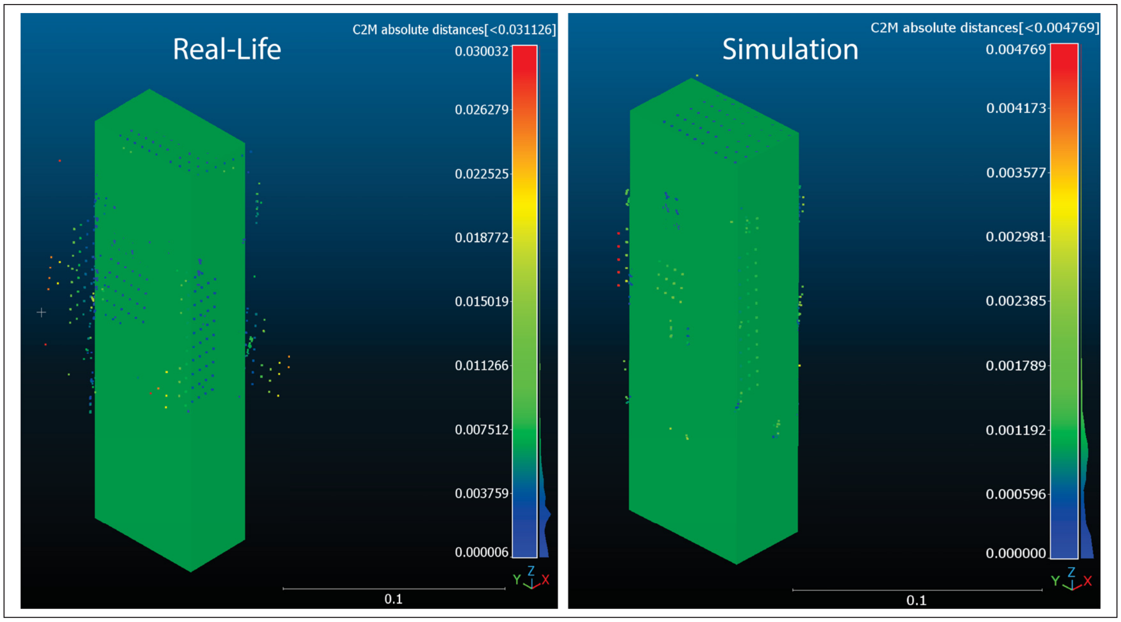


Figure 5.7 Rectangular prism CAD alignment for both real-life and simulation point clouds before data treatment

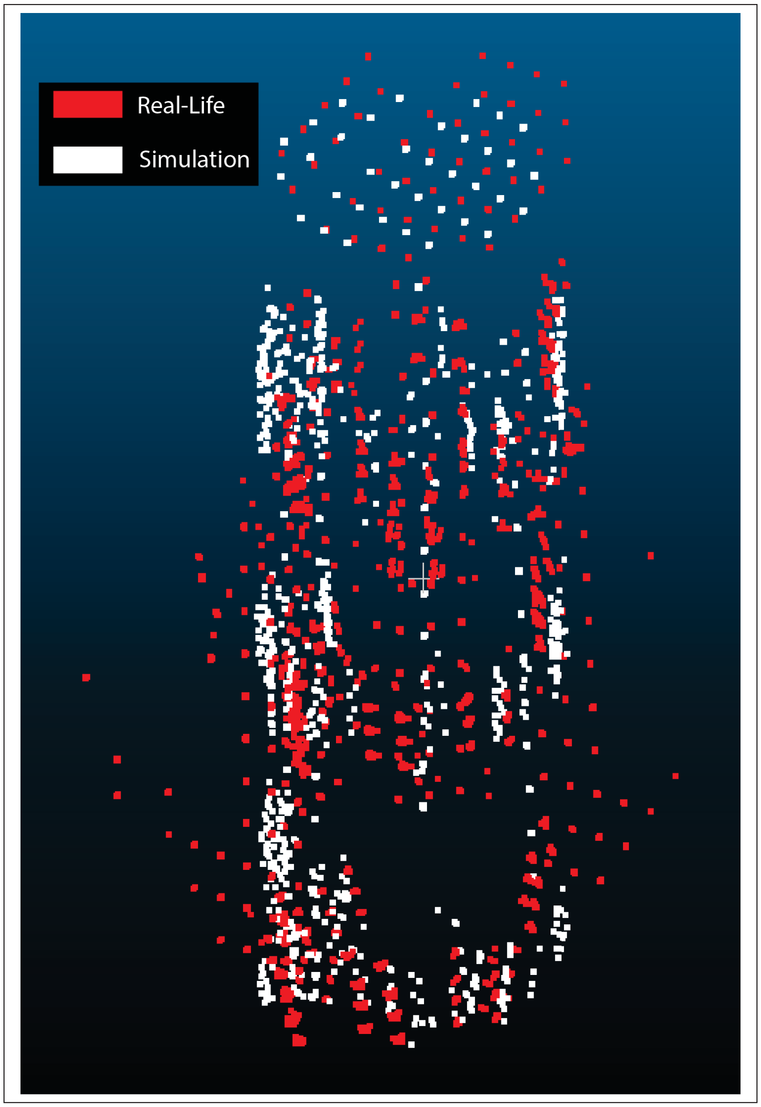


Figure 5.8 Alignment between real-life and simulation point clouds before data treatment for the cylindrical prism

For the alignment, the ICP (*Iterative closest point*) algorithm was used, with 1000 iterations, a final overlap of 100 % and a random sampling limit equal to the number of points in the cloud. As for the cloud to mesh distance calculation, the Octree spatial partitioning method was used with a level of six in this case along with unsigned distances. These alignment and distance calculation methodologies were applied to all of the three testing shapes.

5.6.1 Matching Metrics

In order to evaluate the mesh to point cloud or point cloud to point cloud matching quality, three main metrics were used: RMS (*Root mean square*) distance of the ICP algorithm, mean distance and visualization.

The RMS distance, an essential metric derived from the ICP algorithm, serves as a fundamental measure of alignment accuracy in the realm of point cloud registration. This algorithm iteratively refines the transformation between two datasets (i.e. point clouds and meshes), seeking to minimize the RMS distance between corresponding points. The RMS distance quantifies the root mean square of the Euclidean distances between the paired points that belong to the aligned entities (i.e. point clouds and meshes), encapsulating the quality of spatial congruence achieved. Please note that unless otherwise specified, the RMS distance in until the end of this thesis refers to the one measured by the ICP algorithm.

Moreover, the mean distance is calculated as the average of the distances between each point in a source dataset (such as a point cloud) and the nearest point on a target dataset (such as a mesh or another point cloud). The mean distance provides a measure of the overall proximity between two datasets, regardless of the alignment method used. It offers an assessment of how closely points in one dataset are located to the points in the other dataset.

while both the RMS distance and the mean distance assess the match between datasets, the RMS distance is specific to the accuracy of point cloud registration achieved through the ICP algorithm, whereas the mean distance provides a broader measure of the proximity between datasets, irrespective of the alignment method. These metrics can be complementary, offering different perspectives on the alignment quality and match between datasets.

Finally, it is worth mentioning that blindly following numerical metrics to assess the matching quality in this case is not advisable ans it is necessary to take the visualization of the matching results into account. For example, in some case as it will be shown later when presenting the tabulated results, while its RMS and mean distances are really low, the point cloud being assessed is not semantic and does not provide a descent representation of the grasped object and therefore cannot be considered a good selection.

5.6.2 Data Treatment

A simple observation of the generated point clouds, some of which are presented in figures 5.6 as well as 5.7 and 5.8, reveals that the real-life data are noisy which is inherent in the sensor's operation while in simulation the noise level is much less significant. Moreover, the point clouds generated in real-life have a higher density compared to the ones generated in simulation which can be attributed to contact propagation along the elastic dielectric surface that covers the tactile sensors in real-life. Furthermore, in addition to being visually more compatible with the corresponding CAD models, the simulation point clouds present better RMS and mean distances compared to real-life data and this is depicted in table 5.3.

Table 5.3 A comparison between real-life and simulation points clouds CAD matching results for the different tested shapes and before any data treatment

	Rectangular		**Cylindrical**		**Triangular**	
Metric	Real-life	Simulation	Real-life	Simulation	Real-life	Simulation
ICP RMS (m)	0.00379	0.000805	0.0035	0.000645	0.00512	0.00171
Mean (m)	0.0033	0.00074	0.0033	0.00057	0.00482	0.00153

As we can see both the RMS distance of the ICP algorithm and the mean distance between the point clouds and their corresponding CAD meshes are significantly lower for the simulation data compared to the real-life data and this holds for the three different shapes.

A major point of interest in this project is to generate clean real-life contact point clouds that semantically represent the shapes of the grasped object while also generating synthetic equivalents that are as close to reality as possible. Therefore, starting from this objective along with the previously mentioned observations, three different data treatment techniques were applied to both real-life and synthetic point clouds to minimize the sim-to-real gap. This data treatment is basically improving the CAD matching quality of the real-life point clouds while doing the opposite for the simulation point clouds.

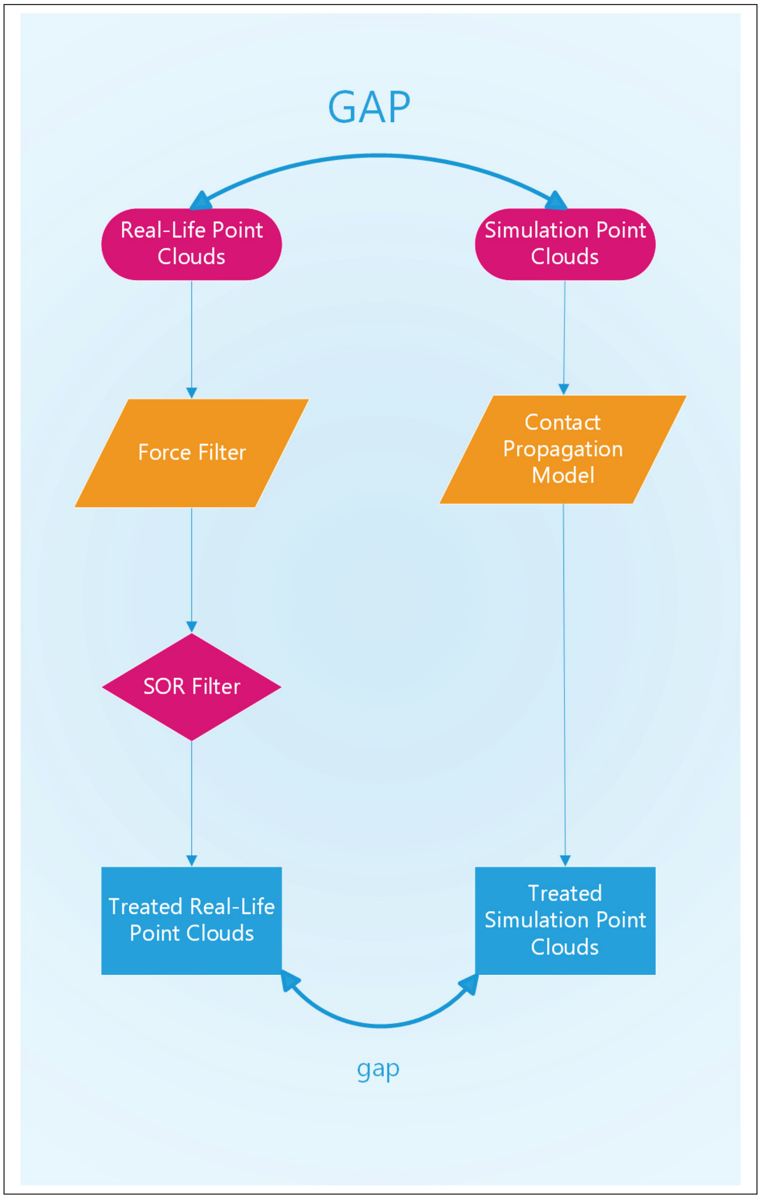
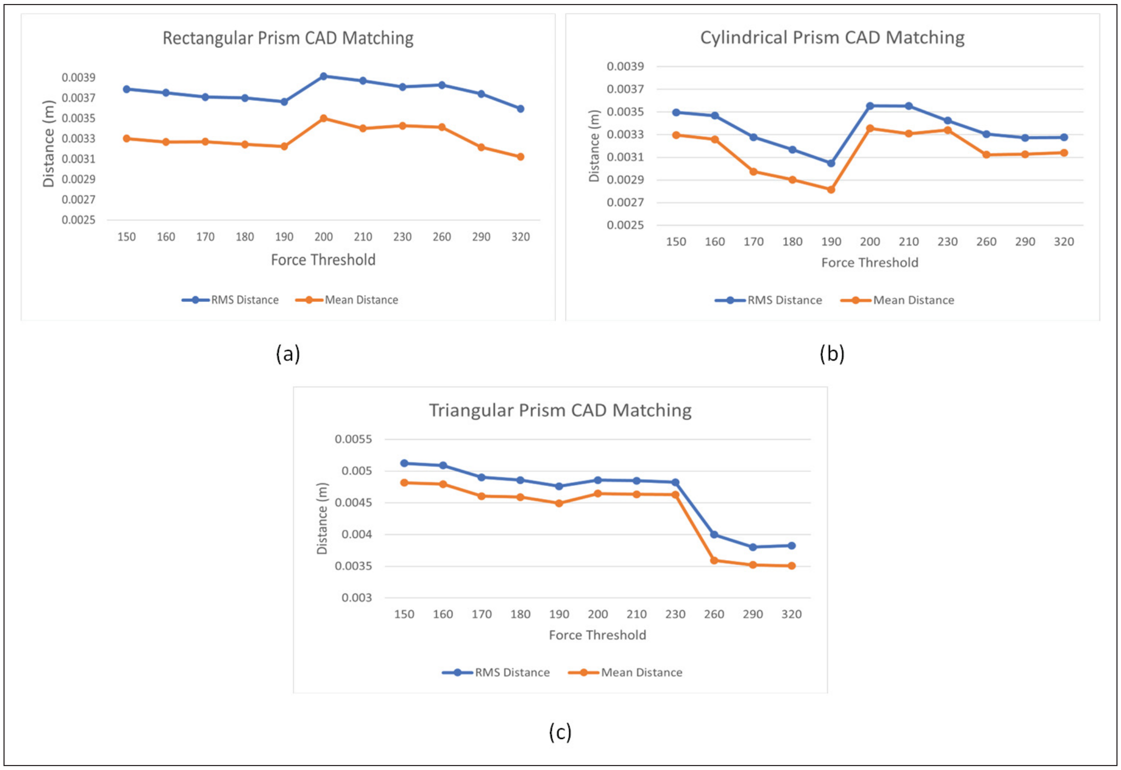

GAP
Real-Life Point Clouds
Simulation Point Clouds
Force Filter
Contact Propagation Model
SOR Filter
Treated Real-Life Point Clouds
Treated Simulation Point Clouds
gap

Figure 5.9 Data treatment for sim-to-real gap minimization

As seen in figure 5.9, as a first step towards reducing the simulation to reality gap, the real-life data were filtered by incrementally increasing the contact threshold, which means that the taxels with less significant contact pressure were considered as noise and were therefore eliminated. As previously mentioned in chapter three, the initial force threshold value was selected to be slightly higher than the sensory signals variations that were introduced by noise in non-contact state.

Nonetheless, the sensor still detects some contact points with low contact forces higher than the set threshold that are actually false contact points and should be eliminated by increasing the threshold level. This filtering process was done while monitoring the variation in both the final RMS distance of the ICP algorithm and the cloud to mesh distance statistics of the cloud to mesh matching. The results for the three different object shapes are presented in tables I-1, I-2 and I-3 of the appendix.
Based on the distance curves presented in figure 5.10, it seems that there is a common trend in the variation of the CAD matching distances as function of the force filtering threshold. For instance, at the beginning both the RMS and mean distances decrease as function of the force threshold until we notice a sudden surge followed by a non-semantic trend. Apparently at first the force filter is effectively removing noisy points from the contact cloud and therefore the quality of the CAD matching which is correlated to the presented distances improves up until reaching a force value of 190 in this case where the semantic data removal becomes more evident which explains the surge in the distance values. Moreover, it is worth noting that despite the fact that the minimum distances are reached at a force threshold much higher than 190, the visualized point clouds at such levels were not semantic and did not properly depict the object's shape. Upon these observations, it seems that all the tactile reading variations from our developed sensor do not represent meaningful contacts and are mostly generated by the noise inherent in the sensor unless they are above the threshold value of 190. As such, the force filter threshold was set to 190 for all of the grasped shapes and this value selection was based on a combination of the RMS and mean distances as well as the resultant point clouds visualized quality.

While contact force filtering improved the simulation to reality gap, it has its own limitations based on our previous discussion. Therefore, to further improve quality of the real-life data and decrease the sim-to-real gap, SOR (*Statistical outlier removal*) filters were applied. SOR filters are powerful tools employed in point cloud processing to enhance data quality by identifying and eliminating outliers, which are data points significantly distant from the majority of points in a dataset. These filters operate based on statistical analysis, by calculating the local mean and standard deviation of point distances within a defined neighborhood around each point in



(a)

(b)

(c)

Figure 5.10 CAD matching evolution with force filter threshold for the rectangular, cylindrical and triangular prism shapes

the point cloud. Points deviating beyond a certain threshold of standard deviations from the local mean are flagged as outliers and subsequently removed or adjusted. The filtering process effectively mitigates the impact of noise, sensor inaccuracies, and anomalies, resulting in cleaner and more reliable point clouds. In this project, the potential of SOR filters was harnessed by experimenting with parameters such as "N" (*Number of points to use for mean distance estimation*) and "nSigma"(*Standard deviation multiplier*) while also monitoring the RMS and mean distances of the CAD matching for each shape, the thing that along with point clouds visualization helped us in identifying the optimal filter settings that yielded the most favorable outcomes, which in this case just like the force filtering scenario were based on a combination of minimal RMS and mean distances as well as a semantic point cloud visualization. The optimal SOR parameters for each shape are presented in table 5.4.

Table 5.4 A comparison among the optimal SOR parameters for the different grasping test shapes

SOR Parameter	Rectangular	Cylindrical	Triangular
N	20	20	716
nSigma	1	2	2

It seems obvious that the optimal SOR parameters varied from one shape to another. These optimal parameters are influenced by many factors like the robotic hand interaction with each shape the thing that affects the distribution of the generated contact points in the 3D space as well as the total number of contact points collected across the different grasps for each shape and the varying geometric surface complexity. For instance in our experiments, while the rectangular shape possesses wider flat faces and therefore has relatively simple surfaces, the cylindrical shape has a more complex curvature and the triangular shape has a higher number of edge to flat areas ratio. These surface complexity differences affect the distribution of the generated contact points for each shape and their respective distances from the ideal ground truth surface.

More specifically, for the optimal filtering results a larger number of points was used for the mean distance estimation for the more complex triangular shape and this can be attributed to the need to consider a larger number of points to capture the irregularities and shape complexities. However, it seems that the rectangular and cylindrical prisms presented less surfaces complexities and in this case a lower lower number of mean estimation points was favorable to capture the simple surfaces and eliminate the anomalies and irregularities. As for the standard deviation multiplier, a higher threshold value was used for the more complex shapes like the curved cylindrical prism and the edgy triangular prism to accommodate the higher irregular contact patterns and deviations inherent in these shapes but for the more consistent contact patterns that are present in the simpler rectangular prism a lower standard deviation multiplier is more favorable to preserve this consistency.

In the end it is worth noting that despite the differences among the SOR parameter values among the different grasped shapes, it is still possible to select one set of parameter values for all the shapes if we were seeking a generalizable rather than an optimal data treatment scheme.

Finally, in order to further decrease the sim-to-real gap, we attempted to increase the density of the simulation point clouds by activating the taxels in the immediate neighbourhood of the already active taxels in the original simulation point clouds. This was done in order to emulate the contact propagation effect through the dielectric of the real-life sensor, which was displayed during the grasping experiments. The CAD matching results of the synthetic point clouds for the different shapes before and after emulating the contact propagation effect are shown in table 5.5. Apparently, the contact propagation emulation has decreased the CAD matching quality of the simulation point clouds. This can be attributed to the fact that the emulated effect has resulted in additional points that deviated from the corresponding ground truth shape surface the thing that increases the noise level and decreases the quality of the simulation point clouds and by that bringing them closer to reality.

Table 5.5 A comparison between the simulation generated tactile point clouds' CAD matching qualities before and after adding the contact propagation effect for all shapes

	Rectangular		**Cylindrical**		**Triangular**	
Metric	Before Contact Propagation	After Contact Propagation	Before Contact Propagation	After Contact Propagation	Before Contact Propagation	After Contact Propagation
ICP RMS (mm)	0.81	1.67	0.65	1.3	1.71	2.21
Mean (mm)	0.74	1.52	0.57	1.2	1.53	1.9

The results of the different data treatment stages that were previously discussed are shown for the rectangular prism object in figure 5.11 and similar results were generated for the other test shapes and depicted in figures I-2 and I-3 of the appendix.

Initially, the real-life point cloud in orange seems to be denser than the simulation point cloud in red and it presents extra outlier points. As previously mentioned, these observations can be attributed to the real-life sensor contact propagation effect and inherent noise respectively. However, as the treatment progresses, the simulation cloud's density increases as seen in yellow after emulating the real-life contact propagation effect and the real-life data gets cleaner with little to no outliers as seen in green after applying the force and SOR filters.

In addition to visualization, the sim-to-real gap progression was monitored by numbers by performing a matching between the point clouds generated in real-life and the corresponding ones that were generated in simulation just like previously shown in figure 5.8. The matching

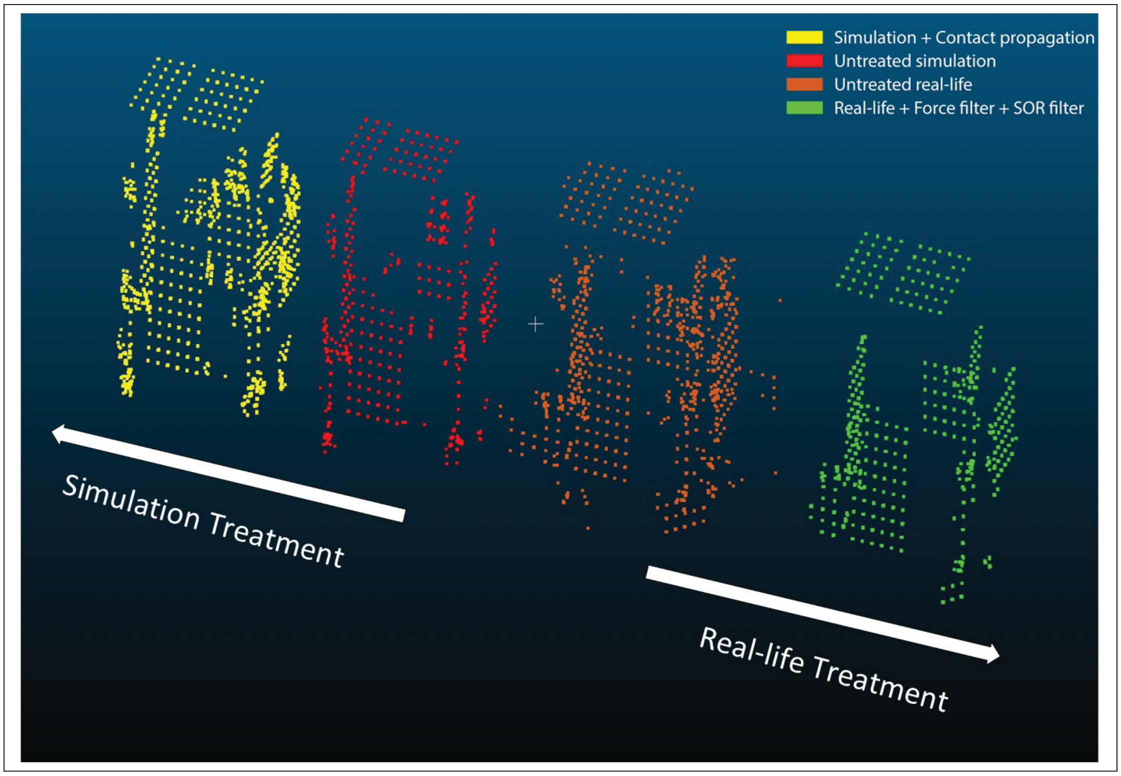
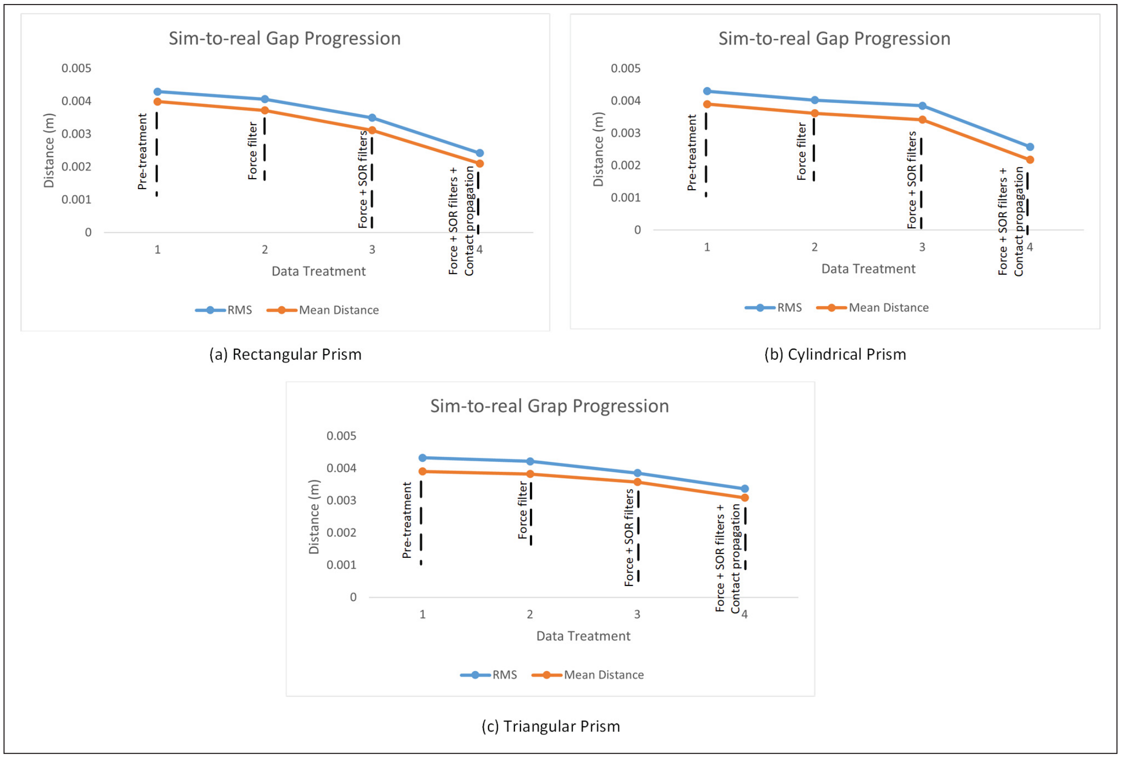


Figure 5.11 Sim-to-real gap progression for the rectangular prism

was performed after each data treatment step and the results were registered in tables I-10, I-11 and I-12 for the three different shapes and the ICP RMS and mean distances progressions among other metrics progressions were extracted and presented in figure 5.12.

A continuous decrease in both distances is observed for all the different shapes and this indicates the added value of each treatment phase in the minimization of the gap between simulation and reality.

Finally further analysis was performed to assess the contribution of each data treatment step to the minimization of the sim-to-real gap and the results for the three object shapes are illustrated in figure 5.13



(a) Rectangular Prism

(b) Cylindrical Prism

(c) Triangular Prism

Figure 5.12 Sim to real gap progression by RMS and mean distances for all the test shapes

It is worth mentioning that in addition to the mean distance, the ICP algorithm's RMS was also monitored to calculate the contribution of each data treatment technique to the sim-to-real gap minimization and the results are presented in the appendix. However, since the same contributions trend was observed only the mean distance based contributions are discussed here. Moreover, the used data treatment techniques were not intended to be applied separately on the real-life and simulation point clouds and as such while it is important to show the progress of there cumulative contribution it would not make sense to measure the contribution of each data treatment technique in isolation as the latter would lead to different and irrelevant results. For example, the contribution of an SOR filter applied on a raw real-life point cloud is different from that of the same filter applied on the force filtered version of the same cloud since these two different filters will interfere with each other and the "outlier" versus "non-outlier "points distribution will change between the point cloud version that went through the force filter and the version that did not.

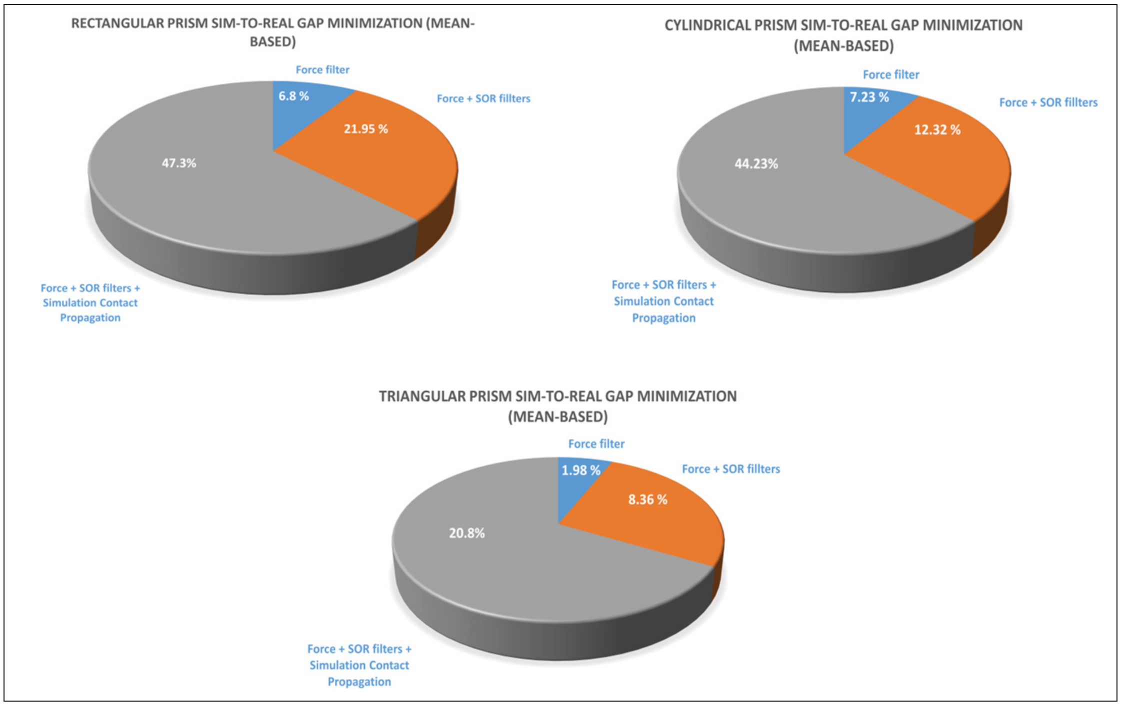


Figure 5.13 Sim to real gap minimization contribution for each data treatment phase for the different shapes: The mean distance for the real-life point cloud to simulation point cloud matching was measured before the data treatment and its decline was monitored after the application of each data treatment technique to register the latter's contribution to the sim-to-real data gap

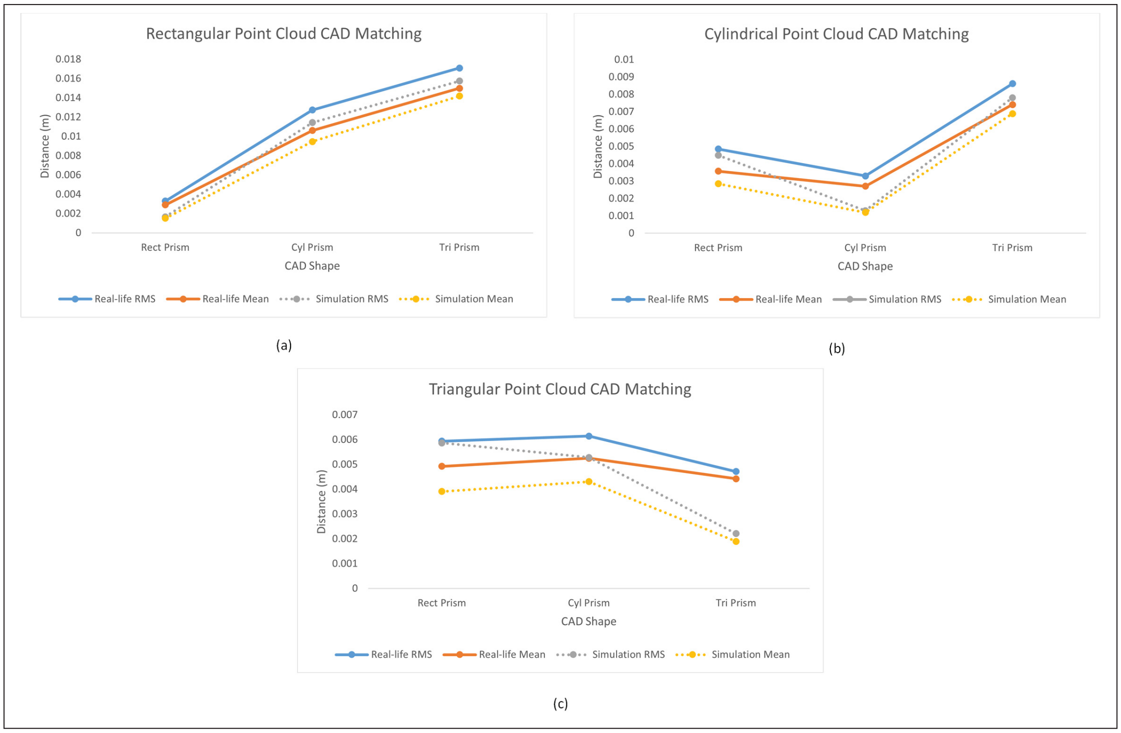
As depicted in the presented diagrams, while all the applied data treatment techniques present an added value towards the sim-to-real gap minimization objective, the force filtering effect has the smallest contribution for all the shapes while the SOR filtering has directly the higher contribution level. This observation reveals that the developed tactile sensor is more susceptible to outlier noise than contact force related noise. This might be attributed to the fact that the majority of the detected contact points have significant contact forces that are higher than the set force filter threshold. However, more noise is coming from more contact points that are generated in isolated locations with respect to the entire point cloud that represents the grasped shapes and this can be explained by the fact that upon contact at specific locations, in addition to activating the corresponding taxels at these locations, the sensor inadvertently activates distant taxels that are unrelated to the immediate tactile interaction. This can be explained by the fact that the electrodes layout of our developed sensor previously discussed in chapter three, was designed as a matrix structure that dictates sharing each electrode among multiple taxels and this can lead to electrical signals interference or a crosstalk among the taxels. While the SOR filter eliminates most of the noise related to this crosstalk, other steps can be taken to mitigate this problem. For instance, in addition to the right selection of the dielectric material as previously discussed in chapter three, the electrodes layout can be further optimized to reduce the impact of the crosstalk among the taxels.

Furthermore, the contribution of the contact propagation effect emulation to the sim-to-real gap minimization seems to be higher than that of the SOR filtering. This can be explained first from the SOR filter side, where one factor that is limiting its contribution might be attributed to the fact that the tactile point clouds that are being filtered are accumulations of multiple point clouds that were collected from multiple grasping attempts. As such, some contact points at certain locations might have been only detected by a small subset of these grasps. As a result this might lead to isolated contact points and low-density contact locations that are genuine but are not consistently present in all the grasping attempts. When the SOR filter is applied, it may remove these legitimate points that are probably detected in simulation and therefore the SOR filter contribution to the sim to real minimization will be negatively affected. This sheds the light on the challenges of combining the data collected from multiple trials or attempts and in this case, in order to allow the SOR filter to retain more of the legitimate or valuable contact points, several data fusion techniques or algorithms can be investigated to combine the data from multiple grasping attempts while having little or no isolated genuine contact points. For example instead of a simple points accumulation, the 3D coordinates of the detected contact points can be averaged across the different attempts of the same grasp which helps in reducing the variations caused by different grasp executions which leads to less isolated genuine contact points occurrences.

Furthermore, the contact propagation effect emulation has the highest contribution to the sim-to-real gap minimization probably because this effect occurs with varying degrees not only across all the sensors covering the Allegro robotic hand used in our experiments but also across all the experiments' grasping executions while the previously discussed force and outlier related noise are more localized and sporadic and may not be as consistently present across all the grasping executions. Therefore adding this effect to the simulation generated tactile point clouds will have a more prevalent effect on the sim to real gap minimization than the force and outlier noise filtering.

Another interesting observation that was made is that the contribution of the force filter in the case of the triangular prism is much lower compared to the rectangular and cylindrical shapes. Since, the triangular prism has smaller dimensions than the other shapes and has a higher number of edges to flat face areas ratio, therefore the same grasping forces that were applied to the other shapes lead to higher pressure distribution and therefore more sensor dielectric deformation which might have probably lead to higher forces readings in the case of the triangular prism. Consequently, the contact forces are shifted to a higher level leaving a small amount of noise to be removed by the force filter. Moreover, the contribution of the contact propagation effect is the lowest for this shape and this can be attributed to the small triangular prism size, leading to fewer contact points between the grasped shape and the Allegro hand and as such the contact propagation effect has fewer opportunities to manifest compared to the larger rectangular and cylindrical prism shapes.

Finally, we were able to came up with a data treatment sequence not only to minimize the sim-to-real gap but also to improve the matching of the real-life and simulation point clouds with their corresponding shapes CAD models. However, we were also interested in showing that matching of the treated point clouds with the CAD models of different shapes yields comparatively worse results. We believe that this further documents the effectiveness of our proposed data treatment plan and the specificity and robustness of the CAD matching approach in correctly identifying the right shape. Our analysis results are shown in figure 5.14.



(a)

(b)

(c)

Figure 5.14 Real-life and simulation point cloud matching with all the test shapes' CAD models: The treated point clouds that yielded the best matching results with their corresponding shape CAD models were used in the analysis and the ICP's RMS and the mean distances were used to track the matching quality

As we can a common trend is shared among the three different shapes, the ICP's RMS and the mean distance of the real-life and simulation point clouds matching is at its minimum with their corresponding shape's CAD model. This means that the the treated tactile point clouds that were generated from both our real-life and simulation experiments not only capture the right geometry of the grasped shape but one can also use them to identify the shapes that were not grasped by relying on the statistical distance metrics.

CONCLUSION

The field of robotics holds immense promise, with applications ranging from manufacturing precision to collaborative human-robot interactions and the navigation of dynamic and complex environments. Object and shape recognition in cluttered surroundings plays a pivotal role in realizing these applications, enabling robots to perform tasks with precision and adaptability. However, the challenges posed by cluttered environments, partial occlusions, and unreliable visual data highlight the need for alternative sensory modalities. This project addresses this need by developing a cost-effective tactile sensor and leveraging the power of simulation to enhance the performance and capabilities of robotic systems.

By recognizing shapes, this project paves the way towards the future recognition of objects and their properties.Moreover, the significance of this project becomes evident when considering the potential real-world scenarios that could be accomplished in the future. For instance in manufacturing, robots that will be equipped with future iterations of our developed tactile sensor could delicately handle fragile or intricate bodies, ensuring quality control and precision. Moreover, in dynamic and unstructured environments, where visual perception can be unreliable, tactile sensing could equip the robots in the future with adaptability, enabling them to navigate and manipulate objects effectively. Recapping the journey undertaken in this thesis, we initiated with a comprehensive literature review, exploring various transduction mechanisms used in tactile sensors and state-of-the-art approaches to tactile sensor simulation and synthetic data generation. We then ventured into the theory and operating principles of the proposed capacitive tactile sensor. The subsequent chapter delved into the selection and setup of a suitable simulator environment. Following this, we meticulously detailed the sensor's design and hardware setup while offering insights into its cost-effectiveness and ease of application.

An important aspect of our research work included the use of the simulations with the aim to generate realistic synthetic tactile data, providing a cost-effective and convenient approach for AI algorithm training. This undertaking aligned with the advantages simulations offer, including flexibility, the introduction of variability and uncertainty for robust algorithm training, and parallel programming for efficiency gains.

In this research endeavor, we have achieved significant milestones. Our primary accomplishments include the design of a cost-effective tactile sensor capable of precisely detecting the contact force and location, accommodating up to four contact points per cm^2, and operating at a sampling rate of six Hz. Notably, we successfully implemented this sensor across the entire Allegro robotic hand, ensuring comprehensive coverage.

Our rigorous validation process involved conducting real-life grasping experiments on three distinct primitive shapes and generating tactile point clouds to represent these shapes using our developed tactile sensor. Furthermore, we extended our achievements into the realm of simulation, faithfully replicating the sensor's functionality and meticulously recreating the previously mentioned experiments within a simulation environment and generating realistic synthetic tactile point clouds. Subsequently, we employed a meticulously devised data treatment plan that refined the synthetic tactile point clouds, aligning them closely with the real-world counterparts and therefore resulting in the minimization of the sim-to-real disparity.

Perhaps most notably, our research has demonstrated the practical utility of these treated tactile point clouds to accurately identify the specific shape that was grasped during their generation. We believe that our work holds promise for a wide range of future applications in the field of robotics manipulation and object and shape recognition. Most notably, our developed simulation framework can be used in the future to effectively generate large sets of synthetic tactile point clouds to train AI models for shape and potentially object recognition tasks. However, this research project acknowledges its limitations and challenges on both the hardware and simulation levels that are worthy of mentioning.

On the hardware level, the dielectric described in chapter 4 was encapsulated on top of the FPCBs using adhesive tape as part of the current design. While this approach provided a relatively reliable support, it comes with some imperfections. To be specific, the adhesive tape layer had to be changed from time to time to ensure a reliable adhesion of the dielectric layer against the FPCBs and therefore generating more reliable and repeatable tactile data. An alternative approach that is worth investigating, is to cast the polyurethane material, from which the dielectric was fabricated, directly on the FPCBs. This is believed to provide a more reliable and a more consistent long term adhesion, which enhances both the quality and repeatability of the generated tactile data in the real-life experiments. Another challenge that was faced in this project involved the vulnerable connectivity among the developed network of sensors that was previously described in chapter 4. This problem was caused by the fragile inter-PCBs connections that were used for both power and communication delivery among the interconnecting sensors. In order to solve this problem, more robust connections should be used and the developed PCBs should be redesigned to accommodate such connections and it is worth mentioning that the connections design employed in the Allegro hand can be used as an inspiration.

On the simulation level, while a conic microstructure was used to represent the dielectric layer on top of the FPCBs, optimally speaking a simulation model should be created for this layer to replicate its elastic behaviour and accurately predict the contact behaviour of the real-life sensor upon interaction with external objects. Unfortunately, this was one of the limitations of using a relatively computationally cheap solid bodies simulator. Another limitation that was noticed in the context of our conducted experiments was the simulator's inability to accurately represent of the real-life interaction between the Allegro hand and the grasped objects. More specifically, upon contact the surfaces deformations of the colliding bodies should be simulated to reflect the grasped object's tilting and the workstation's table surface deformation during the grasping experiments. Moreover, the interference between the colliding objects should be better simulated to allow minimal penetration. These specifications and requirements can be used to guide a future selection of an alternative simulator. Looking ahead, many research avenues can be explored and many improved versions or even different applications can be built on top of the work conducted in this research project. For instance, the FPCBs can be redesigned to activate the proximity sensing capabilities of the sensor and reap this functionality's potential by using it for material detection for example. Also, instead of representing the robotic grasping workstation as a separate simulation entity, a digital twin can be created to synchronize the real-life and the simulation worlds for a more efficient and a more accurate validation of the simulation experiments. Moreover, the developed system can be used for the future application and/or development of new tactile exploration strategies to efficiently identify objects with more complicated shapes and different material types. Finally, after establishing the adequate frameworks, large synthetic datasets can be generated to train sophisticated AI algorithms and test them for object recognition in real-life as a one step closer towards the ultimate goal of granting the robotic manipulator with the ability to rummage in cluttered environments with minimal or no incorporation of computer vision capabilities.

In conclusion, this thesis project not only contributes to the development of a cost-effective tactile sensors and its integration with simulations but also paves the way for future research and advancements, furthering the perception capabilities of robots in diverse real-world scenarios.

APPENDIX I

REAL-LIFE AND SIMULATION POINT CLOUDS ANALYSIS

1. Figures in annexes

1.1 Sim-to-Real Gap Progression

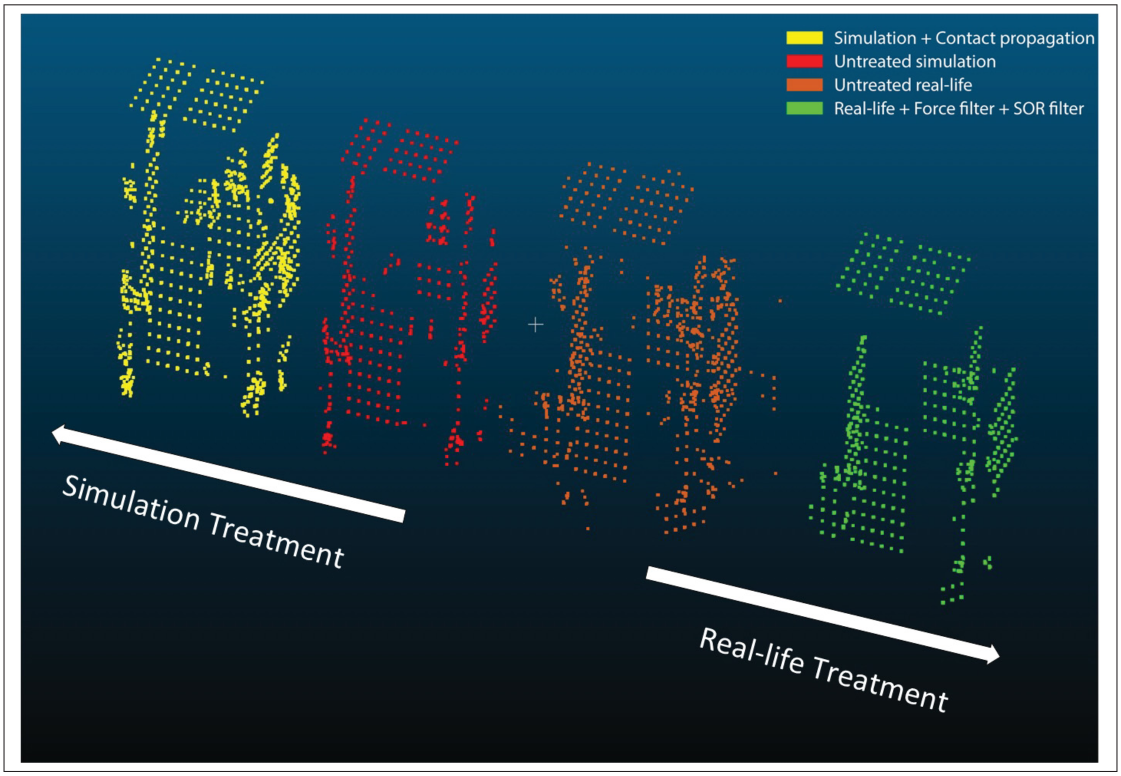


Figure-A I-1 Sim-to-real gap progression for the rectangular prism

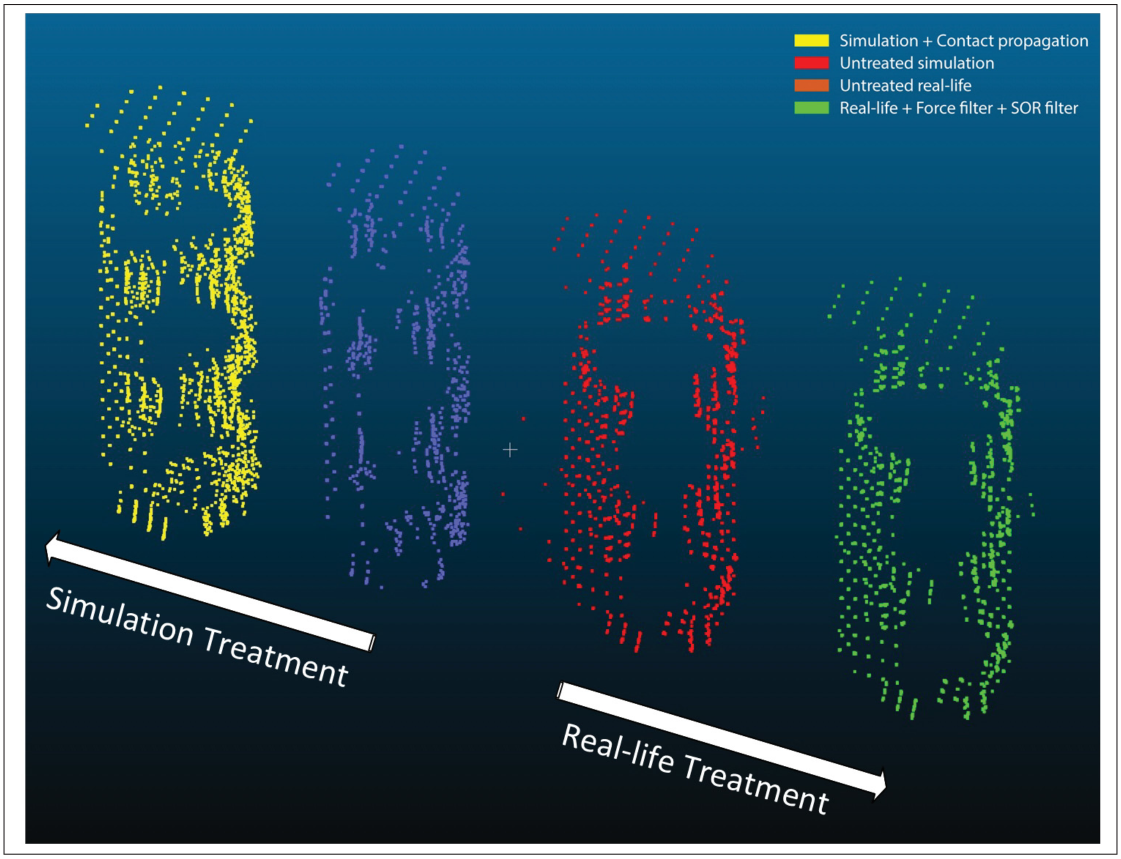


Figure-A I-2 Sim-to-real gap progression for the cylindrical prism

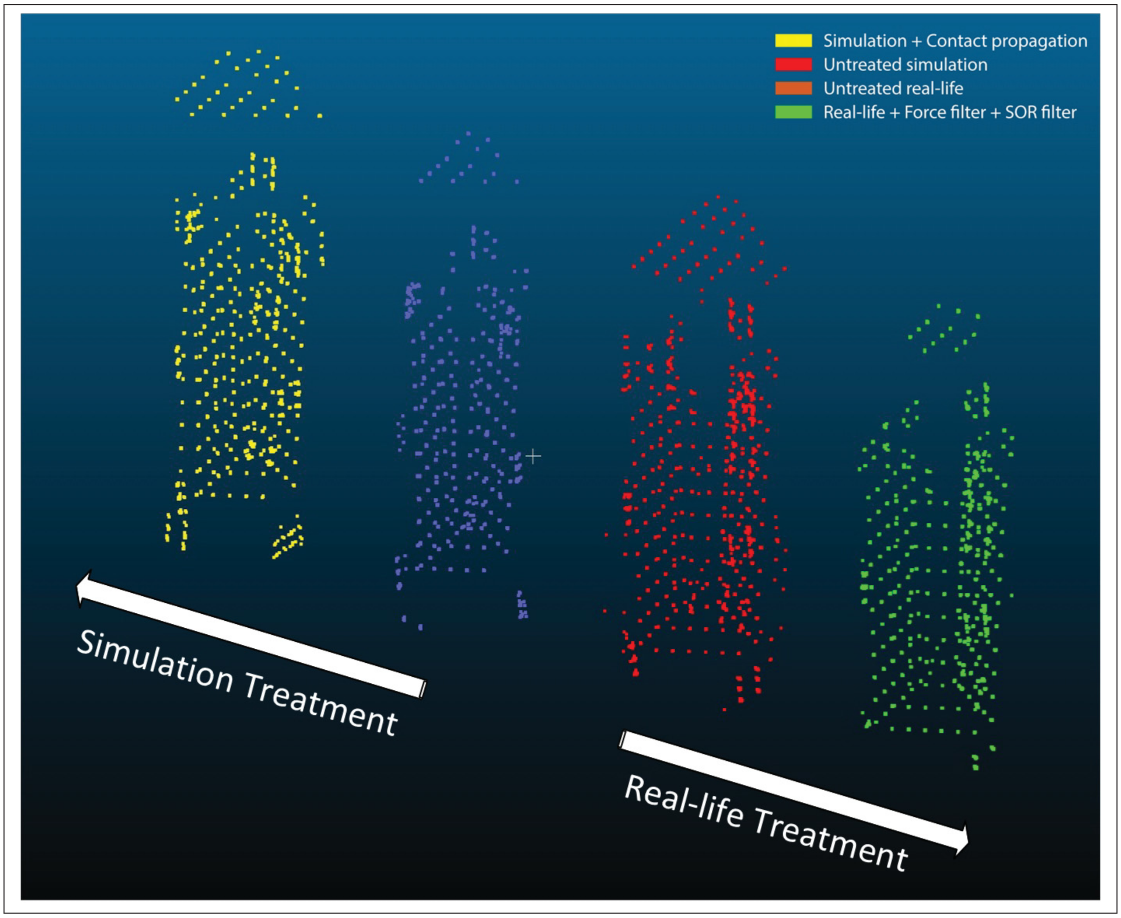


Figure-A I-3 Sim-to-real gap progression for the triangular prism

1.2 Sim-to-Real Gap Minimization Contribution

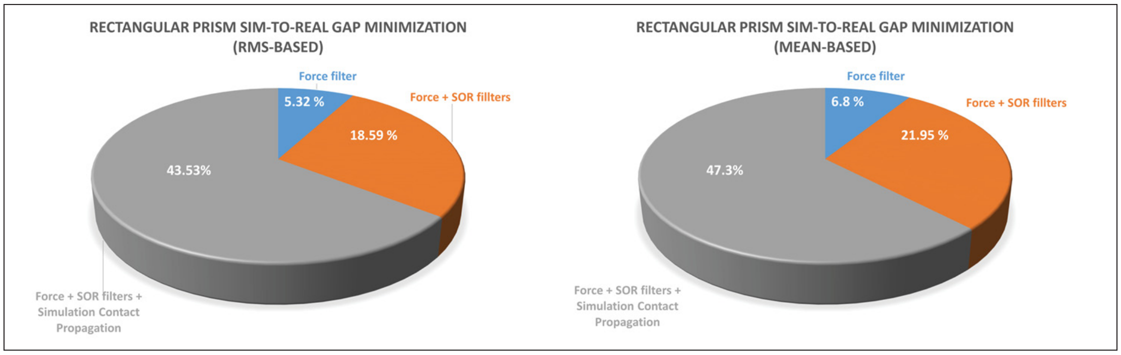


Figure-A I-4 Rectangular prism sim-to-real gap minimization contribution

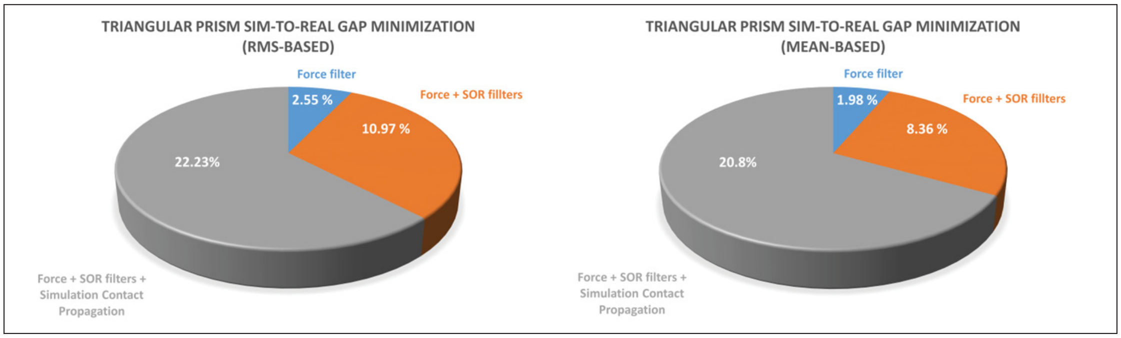


Figure-A I-5 Cylindrical prism sim-to-real gap minimization contribution

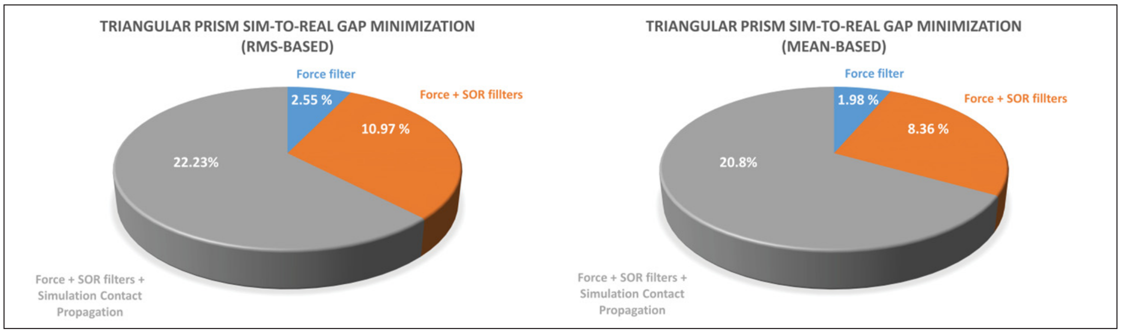


Figure-A I-6 Triangular prism sim-to-real gap minimization contribution

2. Tables in annexes

2.1 Force Filtering Tables

Table-A I-1 Rectangular prism force filtering CAD matching results (Pts = *Points* and Std =*Standard deviation*)

Threshold	**Pts**	**RMS** (m)	**Max** (m)	**Mean** (m)	**Std** (m)	**Filtered out** (%)
150	3504	0.00379	0.02354	0.0033	0.00119	0
160	3387	0.00375	0.02019	0.00327	0.00112	3.339041096
170	3276	0.00371	0.02176	0.00327	0.00113	6.506849315
180	3161	0.0037	0.02176	0.00325	0.00109	9.788812785
190	3069	0.00366	0.02175	0.00323	0.00108	12.41438356
200	2510	0.00392	0.02176	0.0035	0.00128	28.36757991
210	2342	0.00387	0.02175	0.0034	0.00124	33.16210046
230	2111	0.00381	0.02006	0.00343	0.00121	39.75456621
260	1098	0.00383	0.0191	0.00341	0.00119	68.66438356
290	774	0.00374	0.01661	0.00322	0.00114	77.9109589
320	543	0.0036	0.01501	0.00312	0.00111	84.50342466

Table-A I-2 Cylindrical prism force filtering CAD matching results (Pts = *Points* and Std =*Standard deviation*)

Threshold	**Pts**	**RMS** (m)	**Max** (m)	**Mean** (m)	**Std** (m)	**Filtered out (%)**
150	3131	0.0035	0.01938	0.0033	0.00156	0
160	3052	0.00347	0.01938	0.00326	0.00154	2.523155541
170	2949	0.00328	0.01953	0.00297	0.00102	5.812839348
180	2851	0.00317	0.01953	0.0029	0.00087	8.942829767
190	2760	0.00305	0.01953	0.00281	0.00139	11.84924944
200	1902	0.00355	0.01844	0.00336	0.00128	39.25263494
210	1763	0.00355	0.01945	0.00331	0.00117	43.69211115
230	1469	0.00342	0.01679	0.00334	0.00123	53.0820824
260	1122	0.00331	0.01095	0.00312	0.00103	64.16480358
290	851	0.00327	0.01201	0.00313	0.00107	72.82018524
320	610	0.00328	0.01196	0.00314	0.0011	80.51740658

Table-A I-3 Triangular prism force filtering CAD matching results

Threshold	**Pts**	**RMS** (m)	**Max** (m)	**Mean** (m)	**Std** (m)	**Filtered out** (%)
150	3157	0.00512	0.0145	0.00482	0.00168	0
160	3100	0.00509	0.0145	0.0048	0.00163	1.805511562
170	2987	0.0049	0.01235	0.0046	0.00165	5.384859043
180	2874	0.00486	0.01235	0.00459	0.00158	8.964206525
190	2866	0.00476	0.01235	0.00449	0.00158	9.217611657
200	2472	0.00486	0.01199	0.00465	0.00168	21.69781438
210	2462	0.00485	0.012	0.00463	0.00169	22.0145708
230	2225	0.00482	0.01199	0.00463	0.00163	29.52169781
260	458	0.004	0.01011	0.00359	0.00253	85.49255622
290	441	0.0038	0.01012	0.00352	0.00263	86.03104213
320	428	0.00382	0.01013	0.00351	0.00268	86.44282547

2.2 SOR Filtering Tables

Table-A I-4 Rectangular prism SOR filtering CAD matching results (N=*Number of points to use for mean distance estimation* and nSigma=*Standard deviation multiplier*)

N	**nSigma**	**Points**	**RMS** (m)	**Max** (m)	**Mean** (m)	**Standard deviation** (m)	**Filtered out** (%)
20	1	2823	0.0033	0.00764	0.0029	0.00089	8.015640274
	2	2937	0.00338	0.01092	0.00301	0.00105	4.301075269
	3	2981	0.00344	0.01263	0.00302	0.00104	2.867383513
	4	3024	0.00349	0.01263	0.00306	0.00108	1.46627566
2806	1	2444	0.0042	0.01944	0.00385	0.00093	20.36493972
	2	3020	0.00349	0.02275	0.00332	0.0015	1.596611274
	3						
	4						
701	1	2650	0.00343	0.02105	0.00315	0.00109	13.65265559
	2	2962	0.00338	0.02275	0.00311	0.0013	3.48647768
	3	3032	0.00347	0.02275	0.00331	0.00148	1.205604431
	4	3053	0.00351	0.02275	0.00334	0.00149	0.521342457
1403	1	2452	0.00404	0.01746	0.00366	0.00079	20.10426849
	2	2998	0.00345	0.02274	0.00327	0.00146	2.313457152
	3	3065	0.00352	0.02275	0.00335	0.00149	0.130335614
	4						

Table-A I-5 Cylindrical prism SOR filtering CAD matching results (N=*Number of points to use for mean distance estimation*, nSigma=*Standard deviation multiplier*, Pts = *Points* and Std = *Standard deviation*)

N	**nsigma**	**Pts**	**RMS** (m)	Max(m)	**Mean** (m)	**Std** (m)	**Filtered out** (%)
20	1	2483	0.00302	0.01063	0.00282	0.00109	10.03623188
	2	2656	0.003	0.00786	0.00271	0.0009	3.768115942
	3	2707	0.00305	0.01155	0.00283	0.00114	1.920289855
	4	2735	0.003	0.01375	0.00287	0.00122	0.905797101
2759	1	2115	0.00343	0.02183	0.00327	0.00165	23.36956522
	2	2735	0.003	0.02193	0.00271	0.00151	0.905797101
	3						
	4						
690	1	2301	0.00303	0.01256	0.00283	0.00119	16.63043478
	2	2675	0.00311	0.01293	0.00289	0.00129	3.079710145
	3	2754	0.00302	0.02001	0.00271	0.0014	0.217391304
	4						
1380	1	2201	0.00341	0.01357	0.00329	0.0013	20.25362319
	2	2655	0.00305	0.02137	0.00292	0.00152	3.804347826
	3						
	4						

Table-A I-6 Triangular prism SOR filtering CAD matching results (N=*Number of points to use for mean distance estimation* and nSigma=*Standard deviation multiplier*, Pts = *Points* and Std = *Standard deviation*)

N	**nSigma**	**Pts**	**RMS** (m)	Max(m)	**Mean** (m)	**Std** (*m*)	**Filtered out** (%)
20	1	2712	0.00474	0.00958	0.0045	0.0015	5.373342638
	2	2766	0.00475	0.00911	0.0045	0.00152	3.489183531
	3	2817	0.00474	0.00914	0.00448	0.00157	1.70969993
	4	2833	0.00475	0.00914	0.00449	0.00156	1.151430565
2865	1	2439	0.00488	0.00954	0.00467	0.00139	14.89881368
	2	2543	0.00483	0.01025	0.00462	0.00141	11.27006281
	3						
	4						
716	1	2395	0.00479	0.00908	0.00456	0.00148	16.43405443
	2	2656	0.00472	0.01027	0.00442	0.00163	7.327285415
	3	2842	0.00474	0.01136	0.00448	0.00155	0.837404047
	4						
143	1	2501	0.00484	0.00923	0.00462	0.00144	12.73551989
	2	2543	0.00483	0.01025	0.00462	0.00141	11.27006281
	3	2850	0.00475	0.0107	0.00449	0.00156	0.558269365
	4						

2.3 Simulation Data Tables

Table-A I-7 Rectangular prism simulations CAD matching results (Sim = *Simulation*)

	Points	**RMS** (m)	**Max** (m)	**Mean** (m)	**Standard deviation** (m)
Original sim	2092	8.05E – 04	0.0056	0.00074	0.00026
Sim+ contact propagation	3468	0.00167	0.00891	0.00152	0.00077

Table-A I-8 Cylindrical prism simulations CAD matching results

	Points	**RMS** (m)	**Max** (m)	**Mean** (m)	**Standard deviation** (m)
Original sim	1682	6.45E – 04	0.00297	0.00057	0.00027
Sim+contact propagation	3236	0.0013	0.00596	0.0012	0.0004

Table-A I-9 Triangular prism simulations CAD matching results

	Points	**RMS** (m)	**Max**(m)	**Mean** (m)	**Standard deviation** (m)
Original sim	2174	1.71E – 03	0.00608	0.00153	0.00075
Sim+contact propagation	2959	0.00221	0.01011	0.0019	0.00113

2.4 Sim-to-Real Data Comparison Tables

Table-A I-10 Rectangular prism simulation-reality cloud matching results (Std = *Standard deviation* and Pts = *Points*)

	Real-life Pts	**Simulation Pts**	**RMS** (m)	**Max** (m)	**Mean** (m)	**Std** (m)
Pre-treatment	3504	2092	0.00429	0.02995	0.00399	0.00135
Force filter	3069	2092	0.00406	0.0299	0.00372	0.0013
Force+ SOR filters	2823	2092	0.00349	0.01098	0.00312	0.00102
Force+ SOR filters+ contact propagation	2823	3468	0.00242	0.00901	0.0021	0.00084

Table-A I-11 Cylindrical prism simulation-reality cloud matching results(Std = *Standard deviation* and Pts = *Points*)

	Real-life Pts	**Simulation Pts**	**RMS** (m)	**Max** (m)	**Mean** (m)	**Std** (m)
Pre-treatment	3131	1682	0.0043	0.02267	0.0039	0.00196
Force filter	2760	1682	0.00403	0.02282	0.00362	0.00144
Force+ SOR filters	2656	1682	0.00385	0.01253	0.00342	0.00104
Force+ SOR filters+ contact propagation	2656	3236	0.00258	0.00743	0.00218	0.00062

Table-A I-12 Triangular prism simulation-reality cloud matching results(Std = *Standard deviation* and Pts = *Points*)

	Real-life Pts	**Simulation Pts**	**RMS** (m)	**Max** (m)	**Mean** (m)	**Std** (m)
Pre-treatment	3157	2174	0.00433	0.01814	0.0039	0.00187
Force filter	2866	2174	0.00422	0.014357	0.00383	0.00178
Force+ SOR filters	2656	2174	0.00385	0.010434	0.00358	0.00143
Force+ SOR filters+ contact propagation	2656	2971	0.00337	0.009768	0.00309	0.0013

2.5 Point Clouds Matching Results with all the Shapes' CAD Models

Table-A I-13 Real-life rectangular prism point cloud matching with all CAD shapes(N=*Number of points to use for mean distance estimation* , nSigma=*Standard deviation multiplier* and Std = *Standard deviation*)

Real-life Rect-Prism Point Cloud								
CAD Shape	**Threshold**	**N**	**nSigma**	**Points**	**RMS** (m)	**Max** (m)	**Mean** (m)	**Std** (m)
Rect-Prism	190	20	1	2823	0.003296	0.00764	0.002896	0.000889
Cyl-Prism					0.012737	0.02417	0.010608	0.007044
Tri-Prism					0.017091	0.03101	0.014989	0.008211

Table-A I-14 Simulation rectangular prism point cloud matching with all CAD shapes

Simulation Rect-Prism Point Cloud					
CAD Shape	**Points**	**RMS** (m)	**Max** (m)	**Mean** (m)	**Standard deviation** (m)
Rect-Prism	3468	0.001667	0.00891	0.00152	0.00077
Cyl-Prism		0.011431	0.02042	0.00946	0.006421
Tri-Prism		0.015742	0.02789	0.01419	0.006804

Table-A I-15 Real-life cylindrical prism point cloud matching with all CAD shapes(N=*Number of points to use for mean distance estimation*, nSigma=*Standard deviation multiplier* and Std = *Standard deviation*)

Real-life Cyl-Prism Point Cloud								
CAD Shape	**Threshold**	**N**	**nSigma**	**Points**	**RMS** (m)	**Max** (m)	**Mean** (m)	**Std** (m)
Rect-Prism	190	20	2	2656	0.00485	0.0169	0.00358	0.00328
Cyl-Prism					0.0033	0.00786	0.00271	0.000904
Tri-Prism					0.00862	0.02025	0.00741	0.004393

Table-A I-16 Simulation cylindrical prism point cloud matching with all CAD shapes

Simulation Cyl-Prism Point Cloud					
CAD Shape	**Points**	**RMS** (m)	**Max**(m)	**Mean** (m)	**Standard deviation** (m)
Rect-Prism	3236	0.00449	0.0168	0.00285	0.00347
Cyl-Prism		0.0013	0.006	0.0012	0.0004
Tri-Prism		0.00781	0.0166	0.00689	0.00367

Table-A I-17 Real-life triangular prism point cloud matching with all CAD shapes(N=*Number of points to use for mean distance estimation*, nSigma=*Standard deviation multiplier* and Std=*Standard deviation*)

Real-life Tri-Prism Point Cloud								
CAD Shape	**Threshold**	**N**	**nSigma**	**Points**	**RMS** (m)	**Max** (m)	**Mean** (m)	**Std** (m)
Rect-Prism	190	716	2	2656	0.00594	0.0169	0.00492	0.00544
Cyl-Prism					0.00614	0.0118	0.00525	0.00426
Tri-Prism					0.00472	0.0103	0.00442	0.00163

Table-A I-18 Simulation triangular prism point cloud matching with all CAD shapes

Simulation Tri-Prism Point Cloud					
CAD Shapes	**Points**	**RMS** (m)	**Max** (m)	**Mean** (m)	**Standard deviation** (m)
Rect-Prism	2959	0.00587	0.0168	0.00391	0.00437
Cyl-Prism		0.00528	0.0095	0.00431	0.00305
Tri-Prism		0.00221	0.0101	0.0019	0.00113

LIST OF REFERENCES

AG, I. T. (2022). AN64846 Getting Started with CapSense. Infineon Technologies AG.

Agarwal, A., Man, T. & Yuan, W. (2021). Simulation of vision-based tactile sensors using physics based rendering. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–7.

Ahmadi, R., Packirisamy, M., Dargahi, J. & Cecere, R. (2011). Discretely loaded beam-type optical fiber tactile sensor for tissue manipulation and palpation in minimally invasive robotic surgery. *IEEE Sensors Journal*, 12(1), 22–32.

Alagi, H., Heiligl, A., Navarro, S. E., Kroegerl, T. & Hein, B. (2018). Material recognition using a capacitive proximity sensor with flexible spatial resolution. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6284–6290.

Alfadhel, A., Khan, M. A., de Freitas, S. C. & Kosel, J. (2016). Magnetic tactile sensor for braille reading. *IEEE Sensors Journal*, 16(24), 8700–8705.

Ayala, A., Cruz, F., Campos, D., Rubio, R., Fernandes, B. & Dazeley, R. (2020). A comparison of humanoid robot simulators: A quantitative approach. *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pp. 1–6.

Bhirangi, R., Hellebrekers, T., Majidi, C. & Gupta, A. (2021). ReSkin: versatile, replaceable, lasting tactile skins. *arXiv preprint arXiv:2111.00071*.

Castellanos-Ramos, J., Navas-González, R., Fernández, I. & Vidal-Verdú, F. (2015). Insights into the mechanical behaviour of a layered flexible tactile sensor. *Sensors*, 15(10), 25433–25462.

da Silva, J. G., de Carvalho, A. A. & da Silva, D. D. (2002). A strain gauge tactile sensor for finger-mounted applications. *IEEE Transactions on Instrumentation and measurement*, 51(1), 18–22.

Dikhale, S., Patel, K., Dhingra, D., Naramura, I., Hayashi, A., Iba, S. & Jamali, N. (2022). VisuoTactile 6D Pose Estimation of an In-Hand Object Using Vision and Tactile Sensor Data. *IEEE Robotics and Automation Letters*, 7(2), 2148-2155. doi: 10.1109/LRA.2022.3143289.

Ding, Y., Zhang, H. & Thomas, U. (2018). Capacitive proximity sensor skin for contactless material detection. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7179–7184.

Dobrzynska, J. A. & Gijs, M. (2012). Polymer-based flexible capacitive sensor for three-axial force measurements. *Journal of Micromechanics and Microengineering*, 23(1), 015009.

Donlon, E., Dong, S., Liu, M., Li, J., Adelson, E. & Rodriguez, A. (2018). Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1927–1934.

Driess, D., Hennes, D. & Toussaint, M. (2019). Active Multi-Contact Continuous Tactile Exploration with Gaussian Process Differential Entropy. *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7844-7850. doi: 10.1109/ICRA.2019.8793773.

Farley, A., Wang, J. & Marshall, J. A. (2022). How to pick a mobile robot simulator: A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion. *Simulation Modelling Practice and Theory*, 120, 102629.

Habib, A., Ranatunga, I., Shook, K. & Popa, D. O. (2014). SkinSim: A simulation environment for multimodal robot skin. *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1226–1231.

Ivaldi, S., Peters, J., Padois, V. & Nori, F. (2014). Tools for simulating humanoid robot dynamics: a survey based on user feedback. *2014 IEEE-RAS International Conference on Humanoid Robots*, pp. 842–849.

Kappassov, Z., Corrales-Ramon, J.-A. & Perdereau, V. (2020). Simulation of tactile sensing arrays for physical interaction tasks. *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 196–201.

Khadivar, F., Yao, K., Gao, X. & Billard, A. (2023). Online active and dynamic object shape exploration with a multi-fingered robotic hand. *Robotics and Autonomous Systems*, 166, 104461.

Kim, J., Nga Ng, T. & Soo Kim, W. (2012). Highly sensitive tactile sensors integrated with organic transistors. *Applied Physics Letters*, 101(10), 103308.

Kirchner, N., Hordern, D., Liu, D. & Dissanayake, G. (2008). Capacitive sensor for object ranging and material type identification. *Sensors and Actuators A: Physical*, 148(1), 96–104.

Kõiva, R., Schwank, T., Walck, G., Meier, M., Haschke, R. & Ritter, H. (2020). Barometer-based tactile skin for anthropomorphic robot hand. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9821–9826.

Lambeta, M., Chou, P.-W., Tian, S., Yang, B., Maloon, B., Most, V. R., Stroud, D., Santos, R., Byagowi, A., Kammerer, G. et al. (2020). Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3), 3838–3845.

Ledermann, C., Hergenhan, J., Weede, O. & Woern, H. (2012). Combining shape sensor and haptic sensors for highly flexible single port system using Fiber Bragg sensor technology. *Proceedings of 2012 IEEE/ASME 8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, pp. 196–201.

Ledermann, C., Wirges, S., Oertel, D., Mende, M. & Woern, H. (2013). Tactile Sensor on a Magnetic Basis using novel 3D Hall sensor-First prototypes and results. *2013 IEEE 17th International Conference on Intelligent Engineering Systems (INES)*, pp. 55–60.

Li, J., Yin, J., Wee, M. G. V., Chinnappan, A. & Ramakrishna, S. (2023). A Self-Powered Piezoelectric Nanofibrous Membrane as Wearable Tactile Sensor for Human Body Motion Monitoring and Recognition. *Advanced Fiber Materials*, 1–14.

Li, Y., Hu, J., Cao, D., Wang, S., Dasgupta, P. & Liu, H. (2022). Optical-Waveguide Based Tactile Sensing for Surgical Instruments of Minimally Invasive Surgery. *Frontiers in Robotics and AI*, 8, 773166.

Liang, G., Wang, Y., Mei, D., Xi, K. & Chen, Z. (2015). Flexible capacitive tactile sensor array with truncated pyramids as dielectric layer for three-axis force measurement. *Journal of Microelectromechanical systems*, 24(5), 1510–1519.

Mannsfeld, S. C., Tee, B. C., Stoltenberg, R. M., Chen, C. V., Barman, S., Muir, B. V., Sokolov, A. N., Reese, C. & Bao, Z. (2010). Highly sensitive flexible pressure sensors with microstructured rubber dielectric layers. *Nature materials*, 9(10), 859–864.

Maslyczyk, A. (2016). *Développement d'un capteur tactile multimodal.* (Ph.D. thesis, École de technologie supérieure).

Moisio, S., León, B., Korkealaakso, P. & Morales, A. (2013). Model of tactile sensors using soft contacts and its application in robot grasping simulation. *Robotics and Autonomous Systems*, 61(1), 1–12.

Narang, Y., Sundaralingam, B., Macklin, M., Mousavian, A. & Fox, D. (2021). Sim-to-Real for Robotic Tactile Sensing via Physics-Based Simulation and Learned Latent Projections. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6444-6451. doi: 10.1109/ICRA48506.2021.9561969.

Padmanabha, A., Ebert, F., Tian, S., Calandra, R., Finn, C. & Levine, S. (2020). Omnitact: A multi-directional high-resolution touch sensor. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 618–624.

Ridzuan, N. A., Masuda, S. & Miki, N. (2012). Flexible capacitive sensor encapsulating liquids as dielectric with a largely deformable polymer membrane. *Micro & Nano Letters*, 7(12), 1193–1196.

Roberge, E., Fornes, G. & Roberge, J.-P. (2023). StereoTac: a Novel Visuotactile Sensor that Combines Tactile Sensing with 3D Vision. *arXiv preprint arXiv:2303.06542*.

Robotics, W. (2023). Wonik Robotics Official Website. Retrieved on 2023 from: https://www.wonikrobotics.com/.

Robots, U. (2023). Universal Robots Official Website. Retrieved on 2023 from: https://www.universal-robots.com/download/mechanical-e-series/ur5e/robot-step-file-ur5e-e-series/.

Ruth, S. R. A., Feig, V. R., Kim, M.-g., Khan, Y., Phong, J. K. & Bao, Z. (2021). Flexible fringe effect capacitive sensors with simultaneous high-performance contact and non-contact sensing capabilities. *Small Structures*, 2(2), 2000079.

Sferrazza, C., Bi, T. & D'Andrea, R. (2020). Learning the sense of touch in simulation: a sim-to-real strategy for vision-based tactile sensing. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4389–4396.

Si, Z. & Yuan, W. (2022). Taxim: An example-based simulation model for gelsight tactile sensors. *IEEE Robotics and Automation Letters*, 7(2), 2361–2368.

Sicotte-Brisson, J.-C., Bernier, A., Kwiatkowski, J. & Duchaine, V. (2022). Capacitive tactile sensor using mutual capacitance sensing method for increased resolution. *2022 International Conference on Robotics and Automation (ICRA)*, pp. 10788–10794.

Strohmayr, M., Wörn, H. & Hirzinger, G. (2013). The DLR artificial skin step I: Uniting sensitivity and collision tolerance. *2013 IEEE International Conference on Robotics and Automation*, pp. 1012–1018.

Surapaneni, R., Xie, Y., Guo, Q., Young, D. J. & Mastrangelo, C. H. (2012). A high-resolution flexible tactile imager system based on floating comb electrodes. *SENSORS, 2012 IEEE*, pp. 1–4.

Takahashi, H., Nakai, A., Thanh-Vinh, N., Matsumoto, K. & Shimoyama, I. (2013). A triaxial tactile sensor without crosstalk using pairs of piezoresistive beams with sidewall doping. *Sensors and Actuators A: Physical*, 199, 43–48.

Thomasson, R., Roberge, E., Cutkosky, M. R. & Roberge, J.-P. (2022). Going in blind: Object motion classification using distributed tactile sensing for safe reaching in clutter. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1440–1446.

Wang, S., She, Y., Romero, B. & Adelson, E. (2021a). Gelsight wedge: Measuring high-resolution 3d contact geometry with a compact robot finger. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6468–6475.

Wang, S., Lambeta, M., Chou, P.-W. & Calandra, R. (2022). Tacto: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors. *IEEE Robotics and Automation Letters*, 7(2), 3930–3937.

Wang, Y., Lu, Y., Mei, D. & Zhu, L. (2020). Liquid metal-based wearable tactile sensor for both temperature and contact force sensing. *IEEE Sensors Journal*, 21(2), 1694–1703.

Wang, Y., Huang, W., Fang, B., Sun, F. & Li, C. (2021b). Elastic tactile simulation towards tactile-visual perception. *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 2690–2698.

Wattanasarn, S., Noda, K., Matsumoto, K. & Shimoyama, I. (2012). 3D flexible tactile sensor using electromagnetic induction coils. *2012 IEEE 25th International Conference on Micro Electro Mechanical Systems (MEMS)*, pp. 488–491.

Xie, H., Jiang, A., Seneviratne, L. & Althoefer, K. (2012). Pixel-based optical fiber tactile force sensor for robot manipulation. *SENSORS, 2012 IEEE*, pp. 1–4.

Yeo, H. G., Jung, J., Sim, M., Jang, J. E. & Choi, H. (2020). Integrated piezoelectric aln thin film with SU-8/PDMS supporting layer for flexible sensor array. *Sensors*, 20(1), 315.

Yuan, W., Dong, S. & Adelson, E. H. (2017). Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12), 2762.

Zhang, M. M., Kennedy, M. D., Hsieh, M. A. & Daniilidis, K. (2016). A triangle histogram for object classification by tactile sensing. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4931–4938.

Zhang, M. M., Atanasov, N. & Daniilidis, K. (2017). Active end-effector pose selection for tactile object recognition through monte carlo tree search. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3258–3265.

Zhao, X.-F., Wen, X.-H., Sun, P., Zeng, C., Liu, M.-Y., Yang, F., Bi, H., Li, D., Ma, R.-G., Wang, J.-C. et al. (2021). Spider web-like flexible tactile sensor for pressure-strain simultaneous detection. *ACS Applied Materials & Interfaces*, 13(8), 10428–10436.