CHALMERS
UNIVERSITY OF TECHNOLOGY

Hall E-Smart Palm Sensor for Autonomous Home-Based Hand Rehabilitation

Master's thesis in Electrical Engineering

MAGDALENA PUTRYM SKOGBERG
RAHUL RAJAGOPALAN NAIR

DEPARTMENT OF SYSTEMS AND CONTROL
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Hall E-Smart Palm Sensor for Autonomous Home-Based Hand Rehabilitation

MAGDALENA PUTRYM SKOGBERG
RAHUL RAJAGOPALAN NAIR

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Systems and Control
Division of Bionics
Chalmers University of Technology
Gothenburg, Sweden 2025

Hall E-Smart Palm Sensor for Autonomous Home-Based Hand Rehabilitation

MAGDALENA PUTRYM SKOGBERG
RAHUL RAJAGOPALAN NAIR

Supervisor: Emmanuel Dean, Systems and Control, Electrical Engineering
Examiner: Emmanuel Dean, Systems and Control, Electrical Engineering

Master's Thesis 2025
Department of Systems and Control
Division of Bionics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Hall E-smart palm sensor for Autonomous Home-Based Hand Rehabilitation
MAGDALENA PUTRYM SKOGBERG
RAHUL RAJAGOPALAN NAIR
Department of Systems and Control
Chalmers University of Technology

Abstract

Hand rehabilitation is an important process for restoring functionality and quality of life, targeting injuries such as "avocado hand", with tendon or nerve damage. In such applications, estimation of finger forces plays a vital role in understanding tactile interactions and hand function. This thesis presents the design and development of a tactile sensor plate based on Hall-effect sensors, specifically the MLX90393, for detecting and analyzing finger forces and gestures. The primary goal is to design a sensor plate able to perform precise measurement of magnitude and directions of the forces exerted by the fingers, providing real-time feedback to users on 3D force distribution, contact areas, and tactile gestures to enhance interaction analysis and control.

The tactile sensor plate design includes a matrix of 48 Hall-effect sensors, and one Inertial Measurement Unit, embedded within a compact, silicone-coated plate designed for comfort, durability, and ease of cleaning. By collecting sensor data at a sampling rate of 100 Hz, the plate is capable of estimating both normal and shear forces, important for understanding and guiding finger movements. A machine-learning Random Forest model was developed to process sensor data and predict applied force parameters. The results demonstrate the sensor plate's ability to measure force magnitudes and directions with high accuracy, achieving an R^2 value of approximately $96\% - 99\%$ for force prediction in specific test cases.

The research addresses challenges such as hysteresis caused by the silicone layer, calibration for diverse hand sizes, and environmental conditions like magnetic field variations. The system architecture, built on a Raspberry Pi 5 as a main processing unit, opens possibilities for future enhancements, including the integration of additional sensors and improved calibration methods.

This study contributes to the field of tactile sensing and force measurement by offering a precise detection of finger movements. The proposed system can estimate the applied force making a foundation for supporting guided exercises, allowing a better recovery process, and reducing dependence on physiotherapists. It also aligns with sustainable design principles. Future work will focus on enhancing sensor accuracy, refining the graphical user interface, and exploring potential applications in various tactile sensing and human-machine interaction scenarios.

Keywords: Hall-effect sensor, force measurement, machine learning, sensor plate, MLX90393, home-based therapy, decision tree.

Acknowledgments

I would like to express my deepest gratitude to my parents, husband, and children for their constant support, patience, and belief in me throughout this journey. Their encouragement and understanding have been my greatest source of strength.
I am truly grateful to my teacher, Emmanuel Dean, for his dedication, guidance, and support.

Magdalena Putrym Skogberg, Gothenburg, March 2025

I am deeply grateful to my supervisor, Dr. Emmanuel Dean, for his constant mentorship and support throughout this research. His guidance and encouragement have been invaluable in shaping this work. I also extend my appreciation to CASE Lab for providing the necessary resources and facilities that enabled me to conduct my research effectively.
Finally, I would like to thank my family and friends for their unwavering support and motivation. Their encouragement helped me push forward, even during the toughest times.

Rahul Rajagopalan Nair, Gothenburg, March 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

Acronyms

AI	Artificial Intelligence
CNN	Convolutional Neural Network
CS	Chip Select (SPI communication)
DEMUX	Demultiplexer
DNN	Deep Neural Network
GUI	Graphical User Interface
I2C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
LCA	Life Cycle Assessment
ML	Machine Learning
MISO	Master In Slave Out (SPI communication)
MOSI	Master Out Slave In (SPI communication)
PCB	Printed Circuit Board
QT	Qt Application Framework
RF	Random Forest
SCL	Serial Clock Line (I2C communication)
SCLK	Serial Clock (SPI communication)
SDA	Serial Data Line (I2C communication)
SDG	Sustainable Development Goals
SPI	Serial Peripheral Interface

Nomenclature

Below is the nomenclature of parameters, variables, and acronyms used throughout this thesis.

Parameters

B	Magnetic flux density (T)
d	Thickness of the conductor (m)
f_{max}	Maximum switching frequency of the demultiplexer (Hz)
F	Applied force (N)
h_k	Filter coefficient in Savitzky-Golay filtering
I	Current through the conductor (A)
k	Spring constant (stiffness of the material)
M	Window size for moving average filtering
m	Half the window size (total window size $2m + 1$)
n	Charge carrier density (m^{-3})
q	Charge of the carriers (C)
R^2	Coefficient of determination for model accuracy
V_H	Hall voltage generated by the sensor (V)
x	Displacement from the original position due to force (m)

Variables

α_x	Roll angle - Rotation around the X-axis (rad)
α_y	Pitch angle - Rotation around the Y-axis (rad)
F_x	Force component in X-direction (N)
F_y	Force component in Y-direction (N)
F_z	Force component in Z-direction (N)

h_k	Filter coefficients (weights for each point in the window)
P_x	Position coordinate in X (m)
P_y	Position coordinate in Y (m)
P_z	Position coordinate in Z (m)
x_{n+k}	Original data point within the window
y_n	Smoothed value at point n

Contents

List of Figures

List of Tables

1

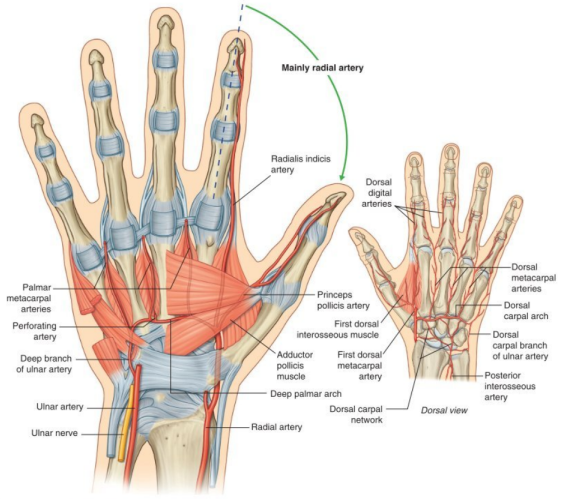
Introduction

This chapter describes the project's goals, explains the research aim, and why the study is important.

1.1 Background

Hand injuries are common in daily life, often resulting in damage to nerves, tendons, or ligaments. One well-known example is the "avocado hand", which occurs when individuals accidentally puncture their non-dominant palm with a sharp tool while cutting an avocado. Improper handling of sharp knives often leads to deep penetrating injuries central palm region of the non-dominant hand, as illustrated in Figure 1.1a.

(a) Avocado hand injury. (Image generated by Gemini, a large language model from Google.)



(b) Hand Anatomy - Nerves and Arteries in the hand. Source: *Gray's Anatomy for Students* [1].

Figure 1.1: Illustration of an avocado hand injury scenario and the anatomical structures that can be affected, including major nerves, arteries, and tendons.

The consequences of such injuries can be serious, affecting both motor and sensory functions of the hand. As shown in Figure 1.1b, the palm contains major nerves (median and ulnar nerves), arteries (radial and ulnar arteries), and muscles with tendons responsible for finger movement. A deep stab injury in this area may lead

to loss of sensation in the affected fingers due to digital nerve damage, impaired grip strength, and dexterity, as flexor tendons are responsible for fine motor movements, excessive bleeding if major blood vessels are compromised, such as the radial or ulnar artery, or even permanent disability or need for surgical intervention in some cases. [1]

Given the complexity and functional importance of the hand, injuries such as "avocado hand" require immediate medical attention, as improper healing can lead to long-term loss of hand function [2]. Tasks may become difficult or even impossible to perform, and some people may no longer be able to work. To restore functionality, proper rehabilitation exercises recommended by healthcare providers are therefore very important. However, for successful recovery, these exercises must be performed within the correct time, with the correct force and technique, adjusted to each individual's needs. Without proper guidance, incorrect movements can worsen the condition. Rehabilitation plays a very important role in restoring function and quality of life after injuries, especially for nerves or musculoskeletal injuries. Without proper rehabilitation, patients are at higher risk of developing long-term impairments, such as chronic pain, reduced mobility, or permanent disability [3]. According to some studies, a well-performed rehabilitation process improves the results and helps patients get back their mobility, strength, and sensor perception in a better way than without rehabilitation. Studies even have shown that early and supervised rehabilitation improves the results compared to unsupervised or incorrect programs, however according to the statistics, only 61% of patients in Sweden receive physiotherapy, and an even smaller percentage - just 7% have access to sensory-focused rehabilitation [3]. This shows that there is a huge gap for those who miss professional rehabilitation, and underscores the need for easily accessible, and effective rehabilitation solutions [3], such as sensor-based technologies. This highlights a gap in access to effective rehabilitation solutions.

To address this gap, a sensor-based tactile force measurement device is proposed, as illustrated in Figure 1.2. This device is still in the early stages of development and cannot yet be considered a rehabilitation tool. However, with further advancements, it has the potential for future applications in rehabilitation.

This device will measure the fingers' force and the direction of the applied forces during tactile interactions. The goal is to provide accurate force estimation and real-time feedback for analyzing finger gestures. This feedback allows users to understand force distribution and adjust their interactions accordingly. Additionally, parameters such as force magnitude and contact patterns could be customized based on specific application needs.

The proposed solution uses Hall-effect sensors. These sensors detect magnetic fields and have been shown to measure normal and shear forces [4] accurately. Measuring shear forces provides insights into the direction of applied forces, helping to track finger movements during tactile interactions. While other sensors, like piezoelectric sensors, may offer higher accuracy [5], the Hall-effect sensor, Melexis MLX90393 was chosen for its suitability for human testing and its ability to measure shear forces [4].

While the accuracy of Hall-effect sensors can be further improved using techniques like Halbach arrays [6], this project focuses on using the MLX90393 sensors in their current configuration. Future projects could explore alternative sensor arrangements

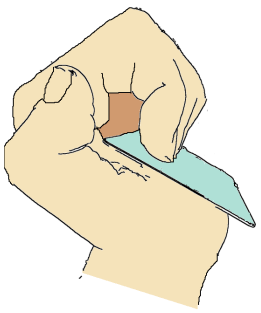


Figure 1.2: Concept of a tactile sensor plate for finger gesture detection. The tactile sensor plate detects finger forces and contact areas, providing data for force estimation and gesture analysis.

or hardware improvements.
In summary, the MLX90393 Hall-effect sensors provide a promising foundation for developing a rehabilitation device in the future. The goal of this project is to create a tactile sensor plate that detects finger gestures accurately during tactile interactions, this technology could make rehabilitation more accessible and effective for patients and healthcare providers.

1.2 Aim

The main aim of this project is to develop a tactile sensor plate using a matrix of MLX90393 sensors. The device will give detailed information about the strength and direction of applied forces, including normal and shear forces. The project has two specific objectives:

1. **Design and Development:** To create a functional tactile sensor plate that operates effectively from an electrotechnical perspective.
2. **Calibration and Testing:** To establish a reliable calibration process that provides accurate measurements for various finger movements and user needs.

The results of this project are expected to contribute to the medical field by offering a tool for improved tracking and support in hand rehabilitation. However, the application to rehabilitation scenarios is beyond the scope of this project.

1.3 Problem

This project focuses on building a Hall-effect sensor-based device for finger gesture detection, providing real-time feedback to users on 3D force distribution and the contact areas. Key challenges include selecting appropriate materials, ensuring sufficient sensor sensitivity, and developing an effective calibration process.
The MLX90393 Hall-effect sensor was selected for its safety and suitability for hu-

man testing. It accurately measures magnetic flux in three axes. This helps in the estimation of both normal and shear forces, which is critical for analyzing finger tactile gestures. A major issue with current rehabilitation methods is the lack of tools that provide accurate, real-time feedback. This often leads to ineffective rehabilitation or even additional injuries. A sensor plate capable of measuring and displaying forces during exercises could help to solve this problem.

The device must be adaptable to different hand sizes and meet individual needs while delivering accurate and consistent results. Practicality and ease of use are also important considerations for daily rehabilitation.

This project focuses on achieving reliable force measurements with the current setup. Future improvements could include enhancing accuracy or enabling the recording of exercise patterns.

In summary, this project aims to develop a Hall-effect tactile sensor plate that measures and provides real-time feedback on applied force parameters, including magnitude and direction.

1.4 Research Question

The following research questions guide this project:

- What accuracy can the sensor plate achieve in measuring the force and direction?
- What resolution is needed to ensure accurate force measurements?
- What design will make the tactile sensor plate easy to clean and maintain?
- What environmental conditions should the tactile sensor plate handle?

These considerations guide the project and guarantee it stays focused on solving the key challenges of finger tactile gesture detection.

1.5 Limitations

This project focuses on developing a prototype for finger gesture detection, with the following limitations for the current stage of the development:

- **User Interface:** A basic interface will be developed for testing; a final user-friendly interface for medical use is beyond the project scope.
- **Power Supply:** Optimizing the power supply is not a focus.
- **Cost:** The project prioritizes functionality over cost-efficiency.
- **Size:** The device will be designed for medium-sized hands only.
- **Inertial Measurement Unit:** Although included in the design, the unit will not be implemented in this phase.
- **Clinical Trials:** The assessment of this tactile sensor will not include clinical setups or real patients, and it will be limited to laboratory assessment in a controlled environment for technical performance evaluation.

Ethical and sustainability aspects

As an engineer contributing to the broader societal context, it is important to assess the ethical and sustainability aspects of the work. In the development of the tactile sensor plate, several considerations align with the United Nations (UN) 17 Sustainable Development Goals (SDGs), reflecting both positive and potential negative impacts on society.

Positive consequences

- **Good Health and Well-being**
  While this device is not currently designed for medical applications, with further development, it could contribute to hand rehabilitation, by supporting healthy living and well-being. Hence, the technology aligns with targets such as reducing consequences of the injuries, ensuring universal access to healthcare, and improving overall health outcomes.
- **Quality Education**
  The project contributes to this goal by ensuring inclusive and equal quality of education. The device's potential can be used in various sectors such as education and training for medical practitioners. This ensures broad access, making it a valuable tool for several purposes. For example, this system can be used in some programs, where individuals, regardless of location or background, can enhance their skills in fields such as mechanics, sports, or playing musical instruments.
- **Reduce income inequality within and among countries**
  The device, upon further development, would provide accessible and effective healthcare options. The device's costs can be optimised by producing it on a large scale. Therefore, the cost of rehabilitation using the device would be affordable to the general public. By offering equal opportunities for recovery regardless of economic, ethnic, or other factors, it promotes social and economic equality.

Negative consequences

- **Responsible Consumption and Production**
  The development and production of electronic components, including those utilised in our tactile force sensor plate, may have consequences regarding responsible consumption and production. This involves considerations such as the sustainable source of materials, energy-efficient manufacturing processes, and minimising waste generation throughout the product lifecycle.
- **Climate Action**
  The electronic industry's activities, including the manufacturing and disposal of electronic components, can contribute to greenhouse gas emissions and climate change. Addressing these impacts requires, for example, minimising energy consumption, adopting renewable energy sources, and an implementing of environmentally friendly manufacturing practices.

- **Life Below Water**
  If electronic waste isn't handled correctly, harmful chemicals and heavy metals can leak into water, harming aquatic life and ecosystems. Proper recycling and disposal are important to stop these dangerous substances from polluting water and destroying marine life.
- **Life on land**
  Similarly, improper disposal of electronic waste can also affect terrestrial ecosystems and biodiversity. Toxic substances leaking from electronic components can contaminate soil and water sources, creating hazardous circumstances for wildlife and human health. Sustainable waste management is necessary to minimize these effects on land-based ecosystems.

[7]

Consideration of Lifecycle Analysis

All these negative consequences may raise concerns regarding the responsible consumption of electronic components. These concerns are connected to the climate change and its effects on life. The electronics industry can help by improving product design, using materials, and recycling better. It's also important to look at the good things eco-friendly practices can bring to the environment.
In considering both the positive and negative consequences of the project, it is important to analyse the product's lifecycle. For example, by performing a comprehensive Life cycle Assessment (LCA), one can evaluate the environmental impacts associated with all stages of a product's life, from raw material extraction to manufacturing, distribution, usage, and termination. Such an analysis will help to identify areas where improvements can be made to minimize negative environmental effects and maximize positive contributions to sustainable development[8].
It's important to think about both the positive and negative effects, especially how they might impact on health, education, industry, and our environment in the long run. The project should follow ethical rules and aim to make a positive difference, supporting the UN's goals for sustainable development.

Structure of the Report

This report is organized into the following chapters:

2. **Current Approaches in Tactile Sensing and Force Measurement for Rehabilitation:** A review of existing technologies and methods for force measurement in rehabilitation devices.
3. **Theory:** A review of the existing theory which supports force estimation methods.
4. **System Architecture:** A detailed description of the design and development process for the tactile sensor plate.
5. **Results:** Analysis of the tactile sensor plate's performance in measuring force and direction, and representing the the Graphical Users Interface.
6. **Discussion:** Evaluation of the results, including limitations and suggestions for future improvements.

2

Current Approaches in Tactile Sensing and Force Measurement for Rehabilitation

This chapter reviews current technologies and methods related to force-sensing devices and explores how machine learning can help process the data collected by these sensors. The focus is on understanding how these tools and methods contribute to the development of rehabilitation systems.

Tactile Sensors in Rehabilitation

Tactile sensing technologies are important in rehabilitation, enabling force measurement for understanding hand movements and patterns. Multiple sensor types, including piezoresistive, capacitive, optical, and Hall-effect sensors have been explored for their potential in medical applications. Piezoelectric sensors offer advantages in capturing the dynamic aspects of touch. Yu et al. developed a flexible piezoelectric tactile sensor, demonstrating its ability to measure dynamic three-axis force. While the sensor achieved good sensitivity and frequency response, an average error of 10.68% in force measurement suggests further refinement is necessary for precise applications [9].

Capacitive sensors, which detect force through changes in capacitance, have also shown promise in tactile sensing for rehabilitation [10]. A recent flexible capacitive tactile sensor, inspired by the micropattern of lotus leaves, demonstrates promising performance with high sensitivity ($0.815kPa^{-1}$), a wide dynamic response range ($0 - 50N$), and a fast response time ($\approx 38ms$) [11]. This sensor also exhibits good reproducibility and can detect bending and stretching forces, suggesting its potential for applications in electronic skins, wearable robotics, and biomedical devices [11]. The ability to measure both normal and shear forces, coupled with the potential for high sensitivity, makes capacitive sensors attractive for applications requiring detailed force mapping. However, challenges remain in achieving long-term stability, hysteresis, durability, and minimizing the influence of environmental factors on measurement accuracy [10] [11]. Another interesting type of sensor is an optical tactile sensor, which detects force through changes in light, and it has also shown promise in tactile sensing for rehabilitation [12]. A recent development in this area is the TacTip family of sensors, which are soft, 3D-printed optical tactile sensors with biomimetic morphologies [12]. These sensors are inspired by the human fingertip and utilize an internal camera to track the movement of pins on the sensor's

surface [12]. The TacTip family of sensors has demonstrated submillimeter accuracy in localization tasks, suggesting their potential for applications in robotic manipulation and human-robot interaction. However, challenges remain in developing robust optical tactile sensors for real-world applications. The TacTip sensors, for example, require complex image processing algorithms to interpret tactile information. Moreover, their long-term durability in real-world settings has not yet been fully established [12]. Another important sensor widely used in medical and robotic applications due to its reliability and precision is the Hall-effect sensor. This sensor measures the magnetic field and can detect both normal (straight down) and shear (sideways) forces. This ability is important for analysing hand movements direction during rehabilitation exercises [13].

Research has demonstrated that Hall-effect sensors are very effective in rehabilitation settings. For example, in one study, tactile arrays using Hall-effect sensors achieved an accuracy of 98.3% for normal forces and 96.8% for shear forces under controlled conditions [13]. These findings confirm the suitability of the MLX90393 sensor model, for measuring forces in rehabilitation, due to its ability to measure magnetic fields in three axes [14]. Kristanto et al. developed a wearable three-axis tactile sensor for fingertips using two MLX90393 sensors, reducing the influence of external magnetic fields. Before their improvements, the external magnetic fields might have caused errors in the force measurement of up to $\pm 4N$. After their improvements, the error due to external magnetic fields was reduced to $\pm 1N$. This is a significant improvement, making the force measurements much more accurate. Li et al., using a single MLX90393 together with a flexible magnetic film, achieved a sensitivity of $13.7mV/N$, and the relationship between force and the output voltage was well-fitted by a linear model ($R^2 > 0.99$) within a force range of 0-10 N [15]. Although these sensors can measure force in multiple directions, are small, and are easy to use, it's still difficult to prevent outside magnetic fields and temperature changes from affecting how accurately they measure [14] [15].

Machine Learning for Force Data Analysis

The data collected by sensors needs to be processed to provide meaningful feedback. Machine learning techniques, such as Convolutional Neural Networks (CNNs) and supervised learning models, are commonly used.

CNNs are particularly effective for analysing complex spatial data, as they can detect patterns and relationships within sensor readings. Hu et al. demonstrated using CNNs for processing data from Hall-effect sensors, achieving high accuracy in predicting force parameters [13]. However, CNNs require significant computational resources, which may limit their use in real-time systems or embedded devices.

Ensemble models, such as Decision Trees and Random Forests, offer an alternative. These models are computationally less demanding and can handle noisy, non-linear data effectively. Farah et al. used Decision Trees for gait phase recognition in orthotic systems, proving their simplicity and efficiency [16]. Similarly, Random Forest models have been employed for sensor calibration, demonstrating their ability to improve data accuracy by handling offset and noise [17]. These models are particularly suitable for embedded systems, where computational resources are limited.

Hysteresis and Non-Linear Data

One challenge in using Hall-effect sensors, particularly in tactile applications, is hysteresis. Hysteresis describes a phenomenon where the sensor's output for a given force depends not only on the current force but also on the history of applied forces. This means that the sensor's reading can differ depending on whether the force is increasing or decreasing, even if the instantaneous force is the same. As Hu et al. [13] note, the silicone layer often used to encapsulate Hall-effect sensors in tactile arrays can contribute significantly to hysteresis. This is because the silicone's deformation under force is not perfectly elastic; it presents some "memory" of past deformations, which affects how the magnetic field is changed and thus how the Hall-effect sensor responds. The hysteresis effect can lead to inaccuracies and delays in force measurements, making it difficult to precisely track dynamic hand movements during rehabilitation exercises. For instance, if a patient is gradually increasing force and then quickly decreasing it, the sensor might lag behind the actual force change due to the hysteresis in the silicone. This can complicate the interpretation of force data and potentially hinder the effectiveness of rehabilitation assessments. Machine learning models, particularly Random Forests as explored by Zou et al. [17] and also mentioned by Hu et al. [13], offer an approach to minimise the impact of hysteresis. By training these models on data that captures the sensor's behaviour under both increasing and decreasing forces, the model can learn to recognize and compensate for the hysteresis effects. The model can essentially learn a mapping that relates the raw sensor output to the actual force, taking into account the history of force application. However, as both studies emphasize, the success of this approach depends on the availability of high-quality, representative training data that adequately captures the hysteresis behaviour of the sensor. Furthermore, precise sensor calibration is important to ensure that the model can accurately generalize to unseen data.

Conclusion

The combination of advanced sensor technologies and machine learning techniques holds considerable promise for enhancing rehabilitation devices. As this chapter has shown, a variety of tactile sensors, including piezoelectric, capacitive, optical, and Hall-effect sensors, offer unique advantages and disadvantages for capturing force information relevant to hand movement analysis. Piezoelectric sensors are very good at capturing dynamic touch, while capacitive sensors offer the potential for detailed force mapping. Optical sensors, like the TacTip family, provide biomimetic approaches to tactile sensing, and Hall-effect sensors, particularly the MLX90393, are valued for their reliability, multi-axis force measurement capabilities, and ease of integration. However, each sensor type also presents challenges, including inaccuracies, sensitivity to environmental factors, and, in the case of Hall-effect sensors, hysteresis.

Machine learning offers powerful tools, address to some of these challenges. CNNs, as demonstrated by Hu et al. [13] can effectively analyse complex spatial data from tactile sensor arrays, enabling accurate force parameter estimation. Random Forest models, as explored by both Hu et al. [13] and Zou et al. [17], provide a computationally efficient alternative for tasks like sensor calibration and hystere-

sis compensation, making them particularly suitable for real-time applications in embedded rehabilitation systems. Utilising the strengths of different sensors and machine learning algorithms, and addressing challenges, such as hysteresis, through proper calibration and data processing will further improve the accuracy and usability of sensor-based rehabilitation systems.

3

Theory

This chapter explains the theoretical principles and methods that form the basis of this project. It focuses on the physical modelling of the system, the application of machine learning techniques, data processing methods, and communication protocols necessary for designing and implementing the tactile sensor plate.

3.1 The Physical Modelling Approach

The physical modelling approach is based on well-established laws of physics that are the foundation for understanding how the system works. These include the Hall effect, Hooke's Law, and the concept of hysteresis, which are all relevant to the functioning of the tactile sensor plate.

3.1.1 Hall Effect

The Hall effect describes what happens in a conductor when a magnetic field **B** is applied perpendicular to the flow of electric current in a conductor, causing the charge carriers to be pushed to one side due to the Lorentz force, which is the force acting on a moving charge in the presence of both an electric and a magnetic field. This creates a voltage difference across the conductor, known as Hall voltage [18]. The Hall voltage is given by the formula:

$$V_H = \frac{IB}{nqd} \tag{3.1}$$

where:

- V_H: Hall voltage,
- I: Current through the conductor,
- B: Magnetic flux density,
- n: Charge carrier density,
- q: Charge of the carriers,
- d: Thickness of the conductor.

Hall-effect sensors measure the Hall voltage to detect changes in magnetic flux \cite{physics}. Since they do not directly measure force, they are typically used to determine the proximity of a magnetic source. By placing a magnet within a deformable medium, such as silicone, the displacement of the magnet in response to an applied force can be measured. This displacement can be related to force using elasticity principles, such as Hooke's Law. This method allows Hall-effect sensors to estimate normal and shear forces in various applications.

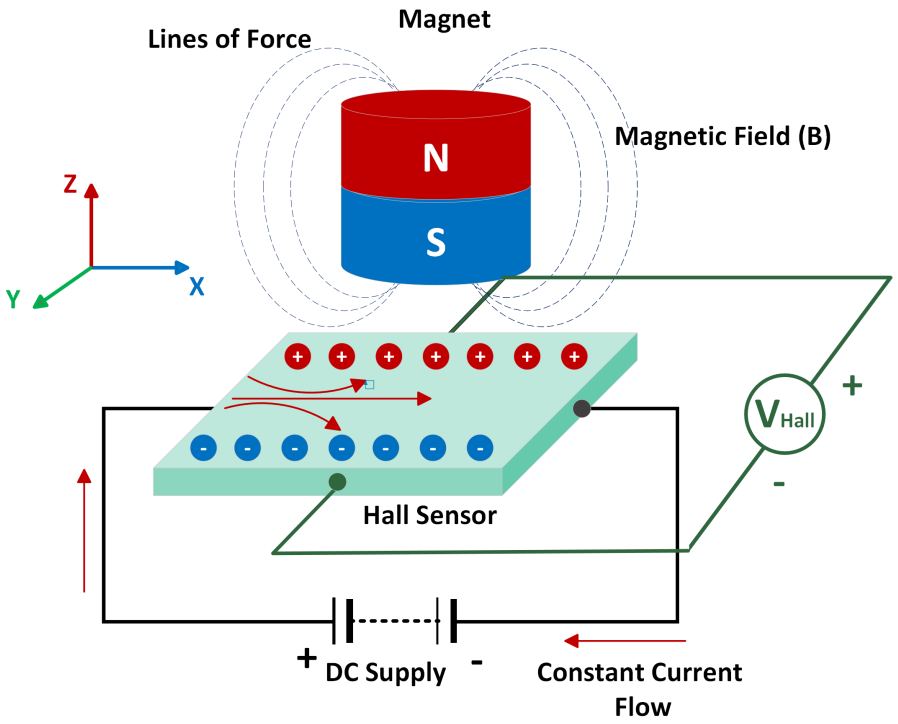


Figure 3.1: Diagram illustrating the working principle of a Hall sensor. The sensor measures the Hall voltage generated when a constant current flows through the sensor in the presence of a magnetic field. This allows the sensor to detect changes in magnetic flux, which can be used to determine parameters like position, speed, and current.

3.1.2 Hooke's Law

Hooke's Law describes the relationship between the force applied to an elastic material and the resulting deformation [19]. The law is expressed as:

$$F = k \cdot x \tag{3.2}$$

where:

- $F \in \mathbb{R}$: Applied force, with dimensions, where $[F] = [\text{N}]$
- $k \in \mathbb{R}^+$: Spring constant (stiffness of the material), $[k] = [\frac{N}{m}]$
- $x \in \mathbb{R}$: Displacement from the original position, with dimensions $[x] = [\text{m}]$

The linear relationship between the force and the distance is valid for an elastic material within its elasticity range, in other words, when the displacement is fully reversible. [19]

3.1.3 Hysteresis

Hysteresis describes how the response of a material depends on its previous state. In elastic materials such as silicone, hysteresis is a delay between the applied force and the resulting deformation. The response differs depending on whether the force

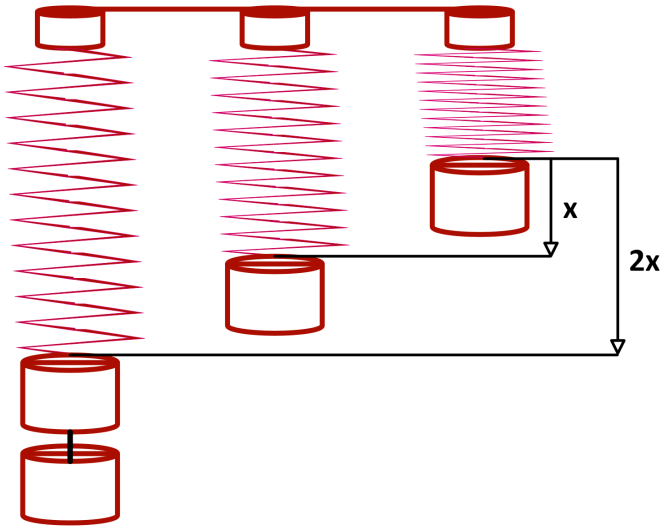


Figure 3.2: Illustration of Hooke's Law. The relationship between the force applied to a spring and its extension. The image shows the linear character of the extension.

is increasing or decreasing, which can introduce errors in force measurements, especially during rapidly changing forces. Figure 3.3 illustrates a typical hysteresis loop, showing how the material's response varies depending on whether the force increases or decreases. [20]

3.2 Machine Learning Approach

The other approach for modelling a system is Machine Learning. This technique is part of a broader field - Artificial Intelligence and is based on algorithms using statistical data and learning from them. The algorithm doesn't take the instructions or rely on predefined rules like a physical approach but makes decisions based on analysing the data and identifying patterns and relationships within the data. To train the machine learning model requires a large dataset, which is why these models have the tendency to improve with time, when more data, or data with improved quality is delivered to the training. The method is especially useful for complex problems, where it is difficult or impossible to model the system based on physical laws due to its complexity or when the relationships between variables are not straightforward and non-linear.[21]
Machine learning models are trained on gathered data. They recognize patterns and on, this basis, adjust the internal parameters to get the desired output based on the provided input. It goes from particular to general relationships between variables, which are later tested on the test dataset - the dataset that was never used for the training [21]. It is a powerful tool for solving problems, where the traditional physical laws are not sufficient or are too complex[17]. Machine learning techniques can be divided into two main categories, based on the type of models and their approach to solving them. These are Supervised Learning Models, which include Decision Trees and Random Forests, and Deep Learning Models, such as Convolutional Neural Networks and Deep Neural Networks.

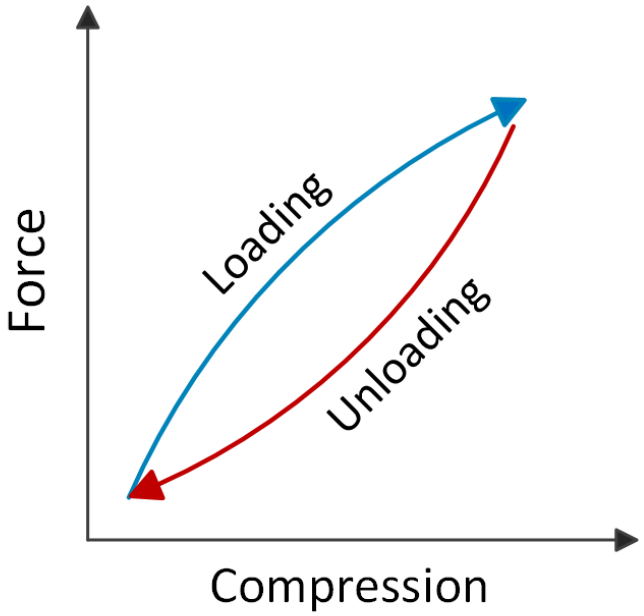


Figure 3.3: Illustration of a hysteresis loop, showing the relationship between applied Force and Compression/Decompression of the silicone.

3.2.1 Supervised Learning Models

Supervised learning models are aimed at handling structured data, interpreting linear and non-linear relationships, and effectively managing noise. These models learn patterns from labelled data, enabling them to make accurate predictions for tasks such as classification and regression [21]. Two common types of supervised learning models are Decision Trees and Random Forests, and both of them belong to the category of tree-based models.

Decision Trees

A decision tree is suitable for tasks such as classification and regression. The model has a tree-like structure and works by dividing the dataset into branches. Each split into a branch is determined by decision rules based on features, where a feature is a measurable characteristic of the analysed data and serves as input to the model. At each internal node, a specific feature is used to make a decision, and each branch represents the outcome of that decision. The leaves of the tree correspond to the final prediction or classification, as illustrated in Figure 3.4.

One of the strengths of a decision tree is its simplicity and interpretability. It is based on a straightforward and transparent process for making predictions, which is especially useful when it is important to understand the reasoning behind a model's output. Another advantage of a decision tree is that it is relatively fast to train and can work effectively with both numerical and categorical data [21].

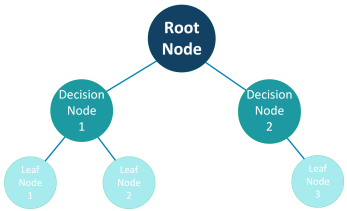


Figure 3.4: Illustration of the Decision Tree

Random Forests

A Random Forest builds on the concept of Decision Trees. Instead of relying on a single tree, a Random Forest uses multiple Decision Trees to make predictions [21], which is why it is also called an ensemble model, see Figure 3.5.

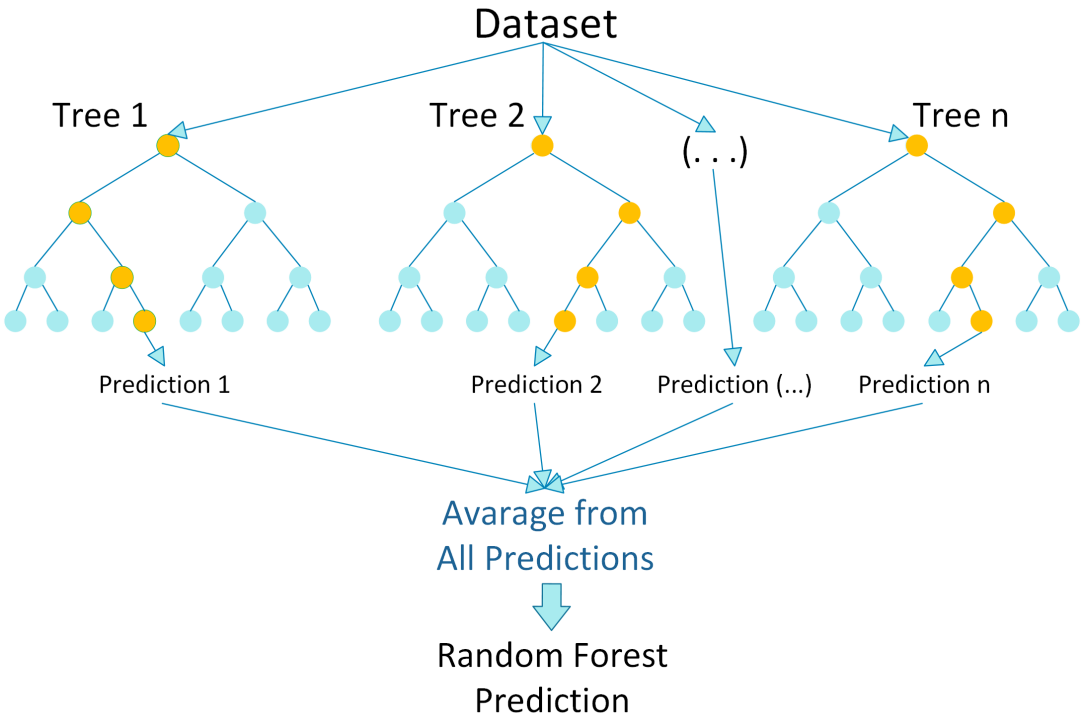


Figure 3.5: Illustration of Random Forest.

Each tree in the forest is trained on a slightly different part of the dataset. This makes the model stronger and more immune to overfitting. For the regression models, the final prediction is a combination of outputs from all the trees, by calculating the average of all tree predictions [21]. This ensemble approach increases the accuracy and is more successful in predicting noisy data or complex datasets than a single Decision Tree. Additionally, they can measure the feature's importance to identify which inputs are relevant and how much impact they have on the output, which helps to sort and eliminate data that doesn't give any information to the model and simplify it.

3.2.2 Deep Learning Models

Deep learning models are aimed at processing high-dimensional data like images or spatial datasets.

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are deep learning models designed for processing data with a grid-like structure, such as images [21]. CNNs work by passing data through several layers, each with a specific function:

- **Convolutional layers:** Use filters (small matrices of numbers) that slide over the input data to find patterns like edges, textures, or shapes.
- **Pooling layers:** Reduce the size of the data by keeping only the most important information. For example, max pooling keeps the largest value in a region, while average pooling calculates the average value.
- **Fully Connected Layers:** The output from pooling layers is sent to fully connected layers, which combine the features to make the final prediction. This process is shown in Figure 3.6.

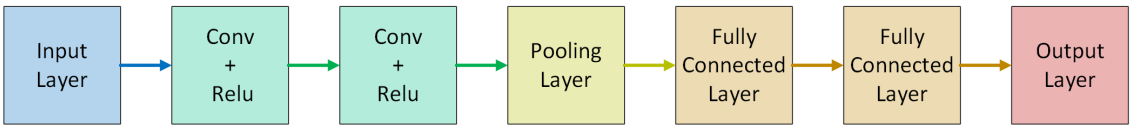


Figure 3.6: Illustration of the Convolutional Neural Network.

CNNs are especially good for finding relationships in spatial problems, where the location and arrangement of features are important. They are often used in tasks such as image recognition to detect objects, patterns, or faces. Another example is in physical modelling, such as analysing temperature distribution across a surface, which is useful in material science to study heat flow or detect defects in materials . The ability to model very complex data increases, however, the complexity of the model often demands more computational resources.

3.2.3 Deep Neural Networks

Another type of machine learning model is Deep Neural Networks (DNNs). These models consist of many layers of connected nodes, called neurons. Each layer processes the input data and step by step turns it into something more abstract. DNNs are very powerful because they can learn complex, non-linear relationships between the input data and the output [21]. However, as CNNs, DNNs need a significant amount of data and computing power to train properly.

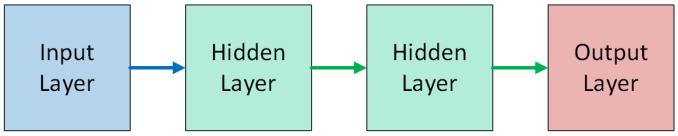


Figure 3.7: Illustration of the Deep Neural Network.

DNNs are flexible models that can be used for many different tasks. However, their complexity can also be a drawback. In some cases, simpler models can give good results with less effort and resources. DNNs are used in tasks such as speech recognition, understanding written or spoken language, and self-driving cars. These are all areas where it is important to find and understand complex patterns in the data.[21]

3.2.4 Using Physical Formulas in Deep Neural Networks

As discussed in [22], physics-informed machine learning incorporates physical knowledge into the learning process, where physical laws are added as constraints in the loss function. The loss function is a measure of how far the model's predictions are from the expected results[21]. By including physical laws in the loss function, the model is penalized when its predictions do not follow these laws. This ensures that predictions respect known principles.
Another method is using physical formulas to compute features that are input to the network. For example, in temperature modelling, Fourier's law can provide derived features such as gradients, improving learning efficiency. Physical knowledge can also be embedded in the DNN structure, such as constraining outputs to follow physical laws or adding custom layers to represent specific formulas.
This combination of learning and physical principles is useful in areas such as material science and fluid dynamics, where it improves accuracy and interpretability, reducing data needs.

3.3 Data Processing and Signal Filtering

Data processing is an important step for ensuring accurate sensor readings. One of the steps of preparing data is filtering. There are many ways in which this can be done. The focus is on one particular method, Savitzky-Golay filtering, which is introduced later in this section.

3.3.1 Savitzky-Golay Filter

The Savitzky-Golay filter smoothens the data and flattens unwanted peaks at the same time keeping the shape and features of the signal, such as peaks and valleys. The basic principle is a window with a fixed size, which applies a polynomial fit over a subset of the data within the window's range, replaces the data points with the predicted values, and minimizes at the same time the least-squares error for the polynomial fit [23]. The window has a fixed size and slides successively over the entire dataset, smoothing it.
The mathematical formula for the Savitzky-Golay filter is [23]:

$$y_n = \sum_{k=-m}^{m} h_k x_{n+k} \tag{3.3}$$

where:

- y_n - Smoothed value at point n,
- x_{n+k} - Original data point within the window,
- h_k - Filter coefficients (weights for each point in the window),
- m - Half the window size (making the total window size $2m + 1$).

The filter's behaviour depends on two parameters:

- **Window's Size:** Determines how many data points are included in the fit. A larger window leads to greater smoothing and, at the same time losing more details. Smaller windows may smooth the signal softer, keeping more details, however leaving more noise.
- **Polynomial Grade:** determines the degree of the polynomial used to the fitting. The higher the polynomial grade, the better the fitting. However, choosing to high grade might lead to unwanted overfitting of the signal's noise.

The Savitzky-Golay filter is popular in fields such as spectroscopy, ECG signal processing, and general data smoothing applications, everywhere it is important to keep the shape of the signal. Proper selection of the window size and polynomial grade for the specific dataset is therefore the key to the optimal performance [24].[23]

3.4 Communication Interfaces

Communication interfaces allow different parts of a system to exchange data. They help the central unit (e.g., a microcontroller) to talk to sensors and other components, making it possible to collect data, process it, and control the system. Two common protocols for this purpose are the Serial Peripheral Interface (SPI) and Inter-Integrated Circuit (I2C).

3.4.1 Serial Peripheral Interface Communication Protocol

The SPI protocol is often used in embedded systems to connect a controller with sensors or other devices. It is simple, fast, and works well for many applications. It can send and receive data at the same time (full-duplex) or in one direction at a time (half-duplex) [25]. In full-duplex mode, SPI uses four wires:

- MOSI (Master Out Slave In): Sends data from the master to the slave.
- MISO (Master In Slave Out): Sends data from the slave to the master.
- SCLK (Serial Clock): Synchronizes the data transfer.
- CS (Chip Select): Selects the device to communicate with.

In half-duplex mode, SPI can use just three wires by combining the MOSI and MISO lines, but this reduces simultaneous communication capability [25].
Figure 3.10 shows the difference between 3-wire and 4-wire SPI configurations.
SPI has several benefits [25]:

- **Fast Data Transfer:** SPI can send data much faster than protocols such as I2C, which is useful for applications that need high-speed data transfer. The transfer speed is usually in the range of 10 - 50 Mhz, but in some cases even up to 100Mhz [26].
- **Simple Design:** SPI is easy to implement and requires less hardware compared to parallel communication methods.

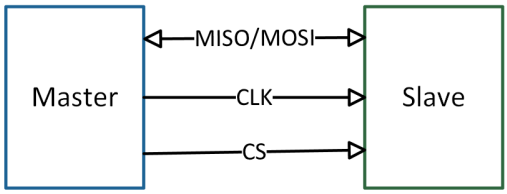


Figure 3.8: SPI 3-Wire Configuration

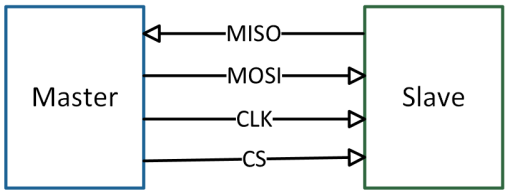


Figure 3.9: SPI 4-Wire Configuration

Figure 3.10: Comparison of SPI Communication Diagrams for 3-Wire and 4-Wire Configurations.

- **Flexible Connections:** SPI allows multiple devices to be connected to one master. Each device is selected using a unique chip select (CS) line.
- **Supports Full-Duplex:** SPI can send and receive data at the same time.

SPI also has some limitations [25]:

- **High Pin Usage:** SPI uses at least four wires in full-duplex mode, and many CS lines, making it more complex to implement if the microcontroller has limited GPIO pins.
- **No Acknowledgment:** SPI does not confirm if the data has been received correctly, which can lead to errors.
- **No Multi-Master Support:** SPI does not handle multiple controllers (masters) easily.
- **Short Distance:** SPI is designed for short distances. On long distances, the signals might become noisy.

3.4.2 I2C Communication Protocol

The Inter-Integrated Circuit communication protocol is another popular interface for communication between a controller and other peripherals. I2C is especially useful in scenarios where multiple devices need to communicate over a short distance using only two wires, which simplifies wiring and reduces complexity. These two lines are the Serial Data Line (SDA) and the Serial Clock Line (SCL). This allows multiple devices to communicate with each other by using unique addresses for each device. This makes I2C suitable for simply connecting multiple devices.
This diagram shows the I2C protocol with a master and three slave devices, indicating the shared SDA and SCL lines 3.11.
I2c protocol has many advantages:

- **Simplicity:** It requires only two wires for communication, which makes the hardware design and wiring simple, especially when multiple devices are involved.
- **Multi-Master, Multi-Slave Capabilities:** It supports multiple master and slave devices on the same bus.
- **Cost-Efficiency:** By reducing the number of connections needed, and slower speed, it makes I2C cost-effective, especially for low-speed peripherals.

However, it has some important disadvantages as well:

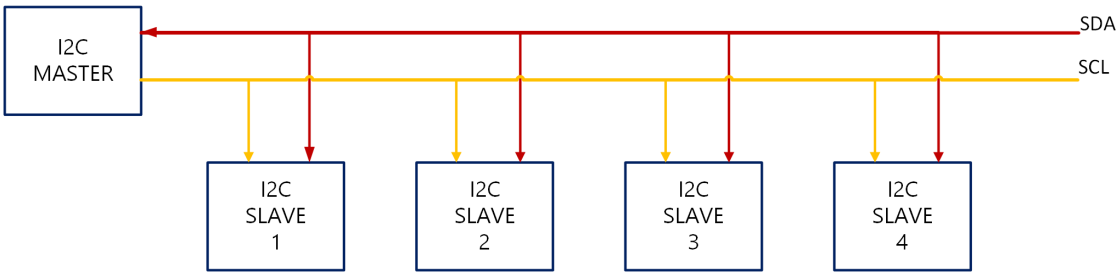


Figure 3.11: I2C communication diagram. The figure illustrates the basic structure of the I2C communication protocol, where multiple slave devices are connected to one master using two shared lines: SDA and SCL. Each device on the bus has its own unique address, allowing the master to communicate with them.

- **Lower Speed:** It operates at slower data rates compared to SPI, the standard mode range is up to 100 kHz, 400 kHz for the fast mode, and up to 3.4 MHz for the high-speed mode.
- **Complexity in Software Implementation:** It requires more complex software management to handle start/stop conditions, addressing, and acknowledgments.
- **Limited Bus Length and Number of Devices:** The bus length and number of devices, that can be connected to it, are limited by the capacitance on the bus.
- **Higher Power Consumption in Active Mode:** It uses more power during communication because of pull-up resistors, especially when running at higher speeds.
- **Risk of Bus Conflicts:** In the case of multiple masters conflicts can happen when two masters try to communicate at the same time. This requires extra software to manage and resolve the issue.[27]

4

System Architecture

This chapter explains the hardware and software components used in developing the tactile sensor plate, as well as their integration to achieve the objectives of this project. The System's Architecture chapter consists of three main parts: electronic components, the physical system design, and the software system. The electronic components section discusses the hardware used in the system, such as the Raspberry Pi, sensors, and any other electronic components. It explains the reasons for choosing these specific components and their technical specifications. The physical System Design section focuses on the physical arrangement and layout of the system. It provides details about the sensor plate design, how the sensors are integrated, and the overall physical structure of the system. The Software System Design section describes the software aspects of the system, including the algorithms used for data processing, force estimation, and user interface. It covers the software architecture and how different software modules interact. Figure 4.1 provides a high-level overview of the system architecture, illustrating the interaction between key modules and the flow of information. The diagram shows the tactile sensor plate's role in data acquisition, the data acquisition and processing, data modeling performed by the Raspberry Pi, and the final presentation of results to the user via the graphical user interface.

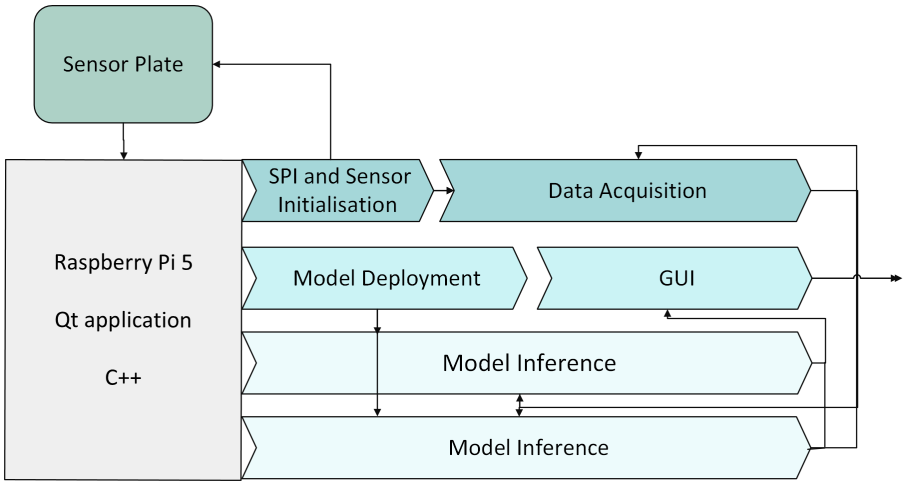


Figure 4.1: System Architecture of the Tactile Sensor Plate System. This diagram illustrates the interaction between key modules in the system, showing the data flow from the tactile sensor plate to the user interface.

As shown in Figure 4.1, the system begins with the tactile sensor plate, which

acquires raw tactile data from the sensors. This data is then transmitted via SPI to the Raspberry Pi 5, where the Data Acquisition and Processing module collects, filters, and prepares the signals for analysis. The Data Modeling module employs machine learning algorithms to predict force parameters based on the processed data. In future implementations, the results from the Data Modeling will be displayed in real-time on the Graphical User Interface, providing users with immediate feedback. The following sections provide an in-depth analysis of each module, starting with a detailed examination of the Tactile Sensor Plate's hardware components and setup, followed by a description of the software algorithms and processing techniques implemented on the Raspberry Pi.

4.1 Electronic Components

The electronic components were selected to guarantee accurate measurements, efficient data processing, and reliable communication between the sensors and the control unit. These components include the sensors, connectors, and the Raspberry Pi computer, which is discussed in detail below.

4.1.1 Raspberry Pi 5

The Raspberry Pi 5 is chosen as the main processor for its speed and promising features, including a 64-bit quad-core ARM Cortex-A76 processor running at 2.4 GHz, making it suitable for handling real-time data and complex tasks. This board was chosen for its sufficient processing power, compact size, and compatibility with various sensors and communication protocols. It also supports the I2C and SPI protocols, which are necessary for data communication between the Raspberry Pi and the connected sensors.[28]

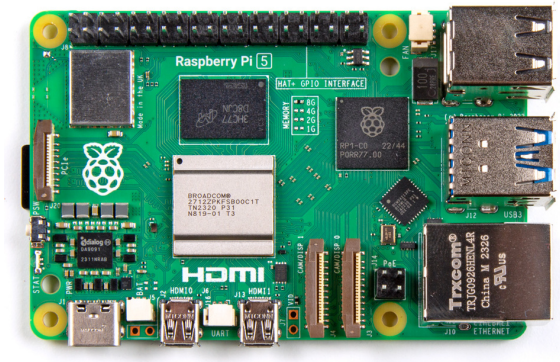


Figure 4.2: Raspberry Pi 5 - a visual representation of the single-board computer. See [29] for details.

4.1.2 Hall-effect Sensor

The Hall-effect sensor, model MLX90393, is a small and precise magnetic sensor designed to measure magnetic fields. It uses Melexis' Triaxis® Hall technology to

measure magnetic flux density in three directions (X, Y, and Z). This enables it to sense 3D positions and detect both normal and shear forces. The sensor's compact design, shown in Figure 4.3, illustrates its suitability for applications with limited space.

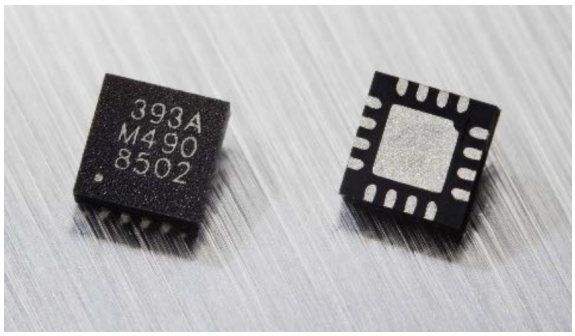


Figure 4.3: MLX90393 sensor diagram. Source: Adapted from Melexis Datasheet [30].

- **High Resolution and Sensitivity:** The MLX90393 has a 16-bit resolution, which makes it very accurate for detecting magnetic fields. It can measure a wide range of magnetic flux densities from 5 mT to 50 mT. The sensor's sensitivity can also be adjusted using programmable gain settings to suit specific needs.
- **Programmable Operating Modes:** It allows its operating modes and sleep times to be adjusted while it is running. Users can balance between performance and power consumption based on their requirements.
- **Compact Design:** It is packaged in a small QFN-16 chip, measuring just 3x3 mm, making it easy to fit into small devices.
- **Popular Communication Interfaces:** The MLX90393 supports both SPI and I2C communication protocols, which are commonly used in electronics. This makes it easy to connect the sensor to different microcontrollers and processors, providing flexibility in system design.
- **Low Power Consumption:** The sensor is designed to use minimal energy, operating on a supply voltage between 2.2V and 3.6V. This makes it suitable for devices with limited power supply.

These features make the MLX90393 a good choice for projects where precise measurements and low energy use are important. Moreover, it is suitable for the tactile sensor plate, because of the small size and measuring accuracy.[30]

4.1.3 Inertial Measurement Unit

The Inertial Measurement Unit (IMU) LSM6DSOQTR[1] is a sensor with a 3-axis accelerometer and a 3-axis gyroscope. It can measure motion with adjustable ranges

[1]The LSM6DSO16ISTR was initially planned for this project due to its integrated Intelligent Sensor Processing Unit, which allows for on-chip data processing. However, since it was out of stock, the LSM6DSOQTR was used instead, as it still meets the project's requirements despite lacking the ISPU feature. [31]

for acceleration (±2/±4/±8/±16 g) and angular rate (±125/±250/±500/±1000/±2000 dps). This IMU tracks motion and is suitable for real-time movement detection [32].

4.1.4 Demultiplexer

The 74HCT154 demultiplexer is an important part of the system, used to manage multiple sensor inputs. It is a 4-to-16-line demultiplexer, which allows control of up to 16 sensors using only four GPIO pins. The 74HCT154 demultiplexer is designed for fast signal switching and efficient sensor selection. The switching speed of the demultiplexer can be estimated using its propagation delay, which represents the time required for a signal to travel through the circuit.
According to the datasheet, the propagation delay from the address input to the output is typically 13 ns, with a maximum value of 35 ns. The transition time, which describes how FAST the output changes state, is around 7 ns. The maximum switching frequency is estimated as follows:

$$f_{\max} \approx \frac{1}{t_{PD}}$$

Using the typical propagation delay of 13 ns, the estimated maximum switching frequency is:

$$f_{\max} \approx \frac{1}{13 \times 10^{-9}} \approx 76.9 \text{ MHz}$$

For the worst-case propagation delay of 35 ns, the switching frequency is reduced to:

$$f_{\max} \approx \frac{1}{35 \times 10^{-9}} \approx 28.6 \text{ MHz}$$

This means that under typical conditions, this demultiplexer can switch at frequencies up to approximately 77 MHz, while under worst-case conditions, the maximum frequency is limited to around 28.6 MHz. The operating frequency may vary depending on supply voltage, temperature, and load capacitance. The practical maximum frequency is generally lower than the theoretical limit.

4.2 Physical System Design

This section explains the design and layout of the tactile sensor plate system, showing how the parts are connected and built.

4.2.1 Tactile Sensor Plate Layout

The tactile sensor plate is designed to hold an array of MLX90393 sensors in a grid pattern. This layout ensures even coverage of the hand's surface, allowing accurate force measurements across different areas. The distance between sensors was chosen to match the common force application points during tactile interactioins.

The schematic in Figure 4.4 shows how the main parts of the tactile sensor plate connect to the Raspberry Pi 5. The tactile sensor plate includes three demultiplexers (DEMUX 1, DEMUX 2, and DEMUX 3) and an inertial measurement unit (IMU), all linked to the Raspberry Pi through a single connector and flat flexible cable. Each

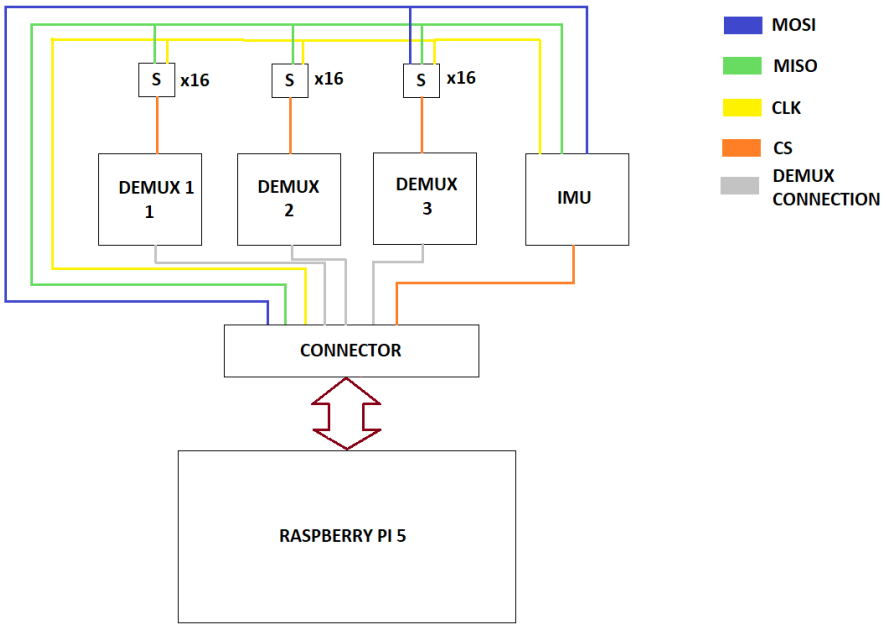


Figure 4.4: Schematic representation of the DEMUX and IMU connections to the Raspberry Pi 5 via a 4-wire SPI interface, including the connections between the DEMUX and the sensors, where S × 16 indicates the presence of 16 sensors.

demultiplexer controls 16 sensors, marked as "S," arranged on the tactile sensor plate, in a matrix array. Demultiplexers control the chip select pin of each sensor (chip select informs the sensor when it is its turn to communicate with the controller), reducing the number of GPIO pins needed to control sensors. The system uses a 4-wire SPI channel for communication, represented by the coloured lines:

- **MOSI (Master Out Slave In, blue):** Sends data from the Raspberry Pi to the sensors and IMU.
- **MISO (Master In Slave Out, green):** Receives data from the sensors and IMU to the Raspberry Pi.
- **CLK (Clock, yellow):** Keeps communication in sync.
- **CS (Chip Select, orange):** Activates specific sensors or the IMU for communication.

The grey lines symbolise the connections between the demultiplexers and Raspberry Pi. The flat flexible cable connects the Raspberry Pi to the tactile sensor plate through the central connector. The short cable with a length of 10 cm minimises signal loss allowing the Raspberry Pi to be mounted on the forearm, making the setup compact and convenient.
Additionally, Figure 4.5 illustrates different layers of the tactile sensor plate, including PCB, silicone layer, magnets, and sensors to help understand the physical design and layering of the device.

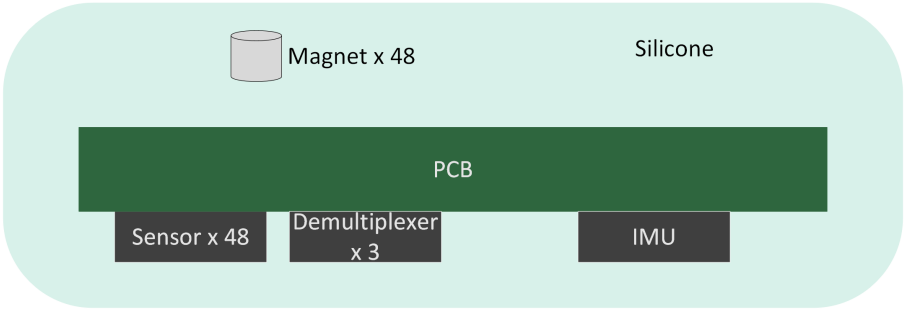
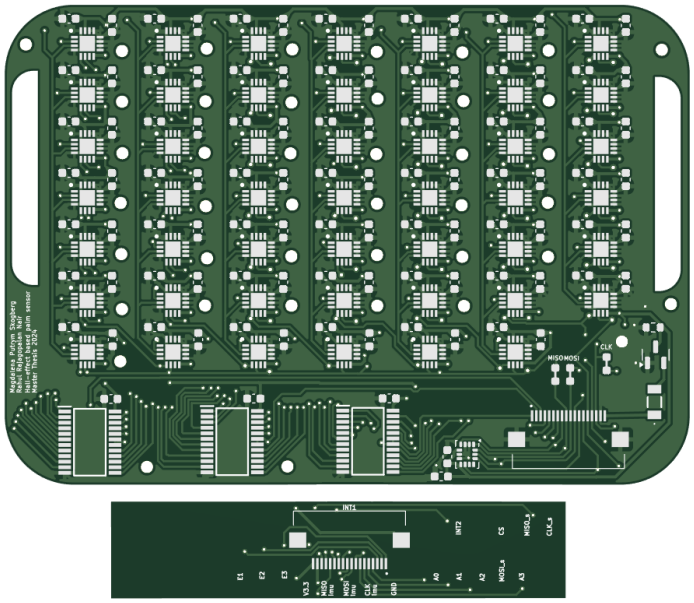


Figure 4.5: Different layers of the tactile sensor board.

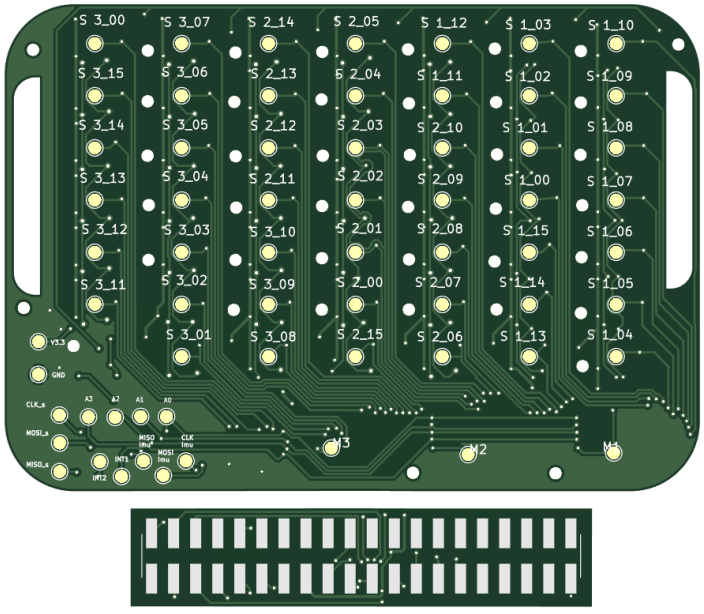
4.2.2 Printed Circuit Board Design

The printed circuit board (PCB), whose schematic is depicted in Figures 4.6a and 4.6b, shows the electrical connection between the tactile sensor plate components. The challenge was to fit as many sensors as possible in the small area constrained by the size of a palm. Moreover, it was also important not to use more than two layers on the PCB to minimize the thickness of the plate. The dimensions of the tactile sensor plate were determined based on handprint measurements collected from approximately 20 people to ensure it fits medium-sized hands. As a result, the plate is designed to hold 48 MLX90393 sensors arranged in a 7x7 grid. The entire plate measures 65 mm vertically and 80 mm horizontally. The sensors are spaced 10 mm apart horizontally and 6 mm apart vertically. The denser vertical spacing helps capture the main forces applied by the fingers during tactile interactions, aligning with the natural movement directions of the fingers. Figure 4.6b illustrates the placement of the sensors and their respective connections to the demultiplexers.

The naming format, such as S 3_00, indicates that the sensor is connected to the third multiplexer and is controlled by its 00 output signal. The labels M1-M3 correspond to the three individual demultiplexers.



(a) Front view of the tactile sensor plate and connectors design, showing the arrangement of the electronic components. (Generated using KiCad)



(b) Back view of the tactile sensor plate and connectors, highlighting the wiring connections and debugging points. (Generated using KiCad)

Figure 4.6: Front and back views of the tactile sensor plate, illustrating the design and wiring of the electronic components.

The back side of the tactile sensor plate contains several debugging points, with

one provided for each sensor, demultiplexer, IMU, and signal line. These points are included, allowing for easy testing and debugging of the system.
Additionally, the tactile sensor plate is designed with multiple 1.5 mm diameter holes. These holes enhance the adherence of the silicone to the plate by allowing the silicone layers on both sides to connect.
The design of the tactile sensor plate also considers other important factors. These include reducing signal noise, keeping signals clean, and managing capacitance. Proper grounding, good trace routing with the proper shape of the lines, spacing between them, and placing capacitors in the right spots to stabilize the power supply by filtering out voltage spikes and noise. These capacitors help keep a stable voltage to the MLX90393 sensors and reduce the impact of power fluctuations and electrical noise on the sensor readings.

Tactile Sensor Plates Power Carrying Capacity

Another important consideration was the power consumption of the tactile sensor plate. The plate integrates several components, each with specific power requirements. The 48 MLX90393 sensors operate in burst mode, with each sensor consuming, in the worst-case scenario, approximately 4 mA [30]. The total current demand for the MLX90393 sensors is calculated as:

$$\text{Total Current (MLX90393)} = 4\,\text{mA} \times 48 = 192.0\,\text{mA}.$$

Additionally, the plate includes one LSM6DSO16 IMU sensor, which has a current consumption of 0.595 mA in the High-Performance Mode[31]. Furthermore, the plate uses three 74HCT154 decoders, each consuming up to 160 µA [33]. The total current demand for the decoders is:

$$\text{Total Current (Decoders)} = 160\,\mu\text{A} \times 3 = 0.48\,\text{mA}.$$

Combining these values, the total theoretical current demand for the tactile sensor plate is:

$$\text{Total Current Demand} = 192.0\,\text{mA} + 0.595\,\text{mA} + 0.48\,\text{mA} \approx 193\,\text{mA}.$$

The current-carrying capacity of the copper traces for both VCC and GND is determined using the IPC-2221 standard, which is a standard that provides design requirements for printed circuit boards [34] and calculated according to the formula:

$$I = k \cdot \Delta T^b \cdot A^c \tag{4.1}$$

where:

- I is the current in amperes (A),
- ΔT is Temperature rise above ambient in $^\circ C$
- A is the cross-sectional area of the trace in mm^2 converted to mil^2 ($1\ \text{mm}^2 = 1550\,\text{mil}^2$),
- k, b, and c are constants from the IPC-2221 standard depending on the environment:,
  - $k = 0.048$, $b = 0.44$, c = 0.725 for external traces.

The VCC and GND traces are made of copper with a thickness of 0.35 mm, and height of 0.035 mm. The current carrying capacity is calculated based on the total cross-sectional area of the traces. The data is taken from the PCB design specifications.
To calculate the total cross-sectional area of the traces, the following formula is used:

$$A_{\text{trace}} = \text{trace width} \times \text{copper thickness} \tag{4.2}$$

- Copper thickness: 0.035 mm
- Trace width: 0.35 mm
- Total cross-sectional area:

$$A_{\text{VCC}} = 0.35\,\text{mm} \times 0.035\,\text{mm} = 0.01225\,\text{mm}^2$$

- A_{VCC} convertion to mil^2:

$$A_{\text{VCC, mil}} = 0.01225\,\text{mm}^2 \times 1550 = 18.99\,\text{mil}^2$$

- Calculation of the current:

$$I_{\text{VCC}} = 0.048 \cdot 10^{0.44} \cdot (18.99)^{0.725} \approx 1.12\,\text{A}$$

Total length of the GND traces is 825.7452 mm and the VCC traces length is 955.2595 mm Traces of the adapter connecting the FFC cables to the Raspberry Pi 5 have the same cross-section area as the main PCB, thus they can carry the same current as the main PCB. The VCC and GND trace are designed to handle more than 5 times higher currents than the operational requirements. This large capacity allows the tactile sensor plate to operate well within the limits under all conditions. The theoretical current demand of approximately 0.2 A has been confirmed by later measurements on a working tactile sensor plate in a burst mode. The power consumption is significantly lower than the calculated capacity, giving a safe margin.
The tactile sensor plate uses an FFC cable to connect to the controller via an adapter PCB. The connection via the cable has one line dedicated to VCC and one to GND. Each FFC cable and connector pair can carry up to 1.0 A per contact, greater than the tactile sensor plate's current carrying demands.

Electrical Protection

The tactile sensor plate is exposed to potential electrical risks during development, testing, and regular usage.
During development and testing the issues may come from faulty connections, such as mistakes in soldering or wiring, which can cause short circuits or unexpected high currents. Using incorrect power supplies or tools during testing might introduce voltage spikes. Additionally, frequent adjustments to the prototype, such as changing configurations or testing new setups, can create temporary electrical issues. Debugging with tools, such as probes can also accidentally short traces or apply unwanted loads.

In everyday use, problems might occur due to user errors, such as plugging in the wrong power supply or using a damaged cable. Overloading the system beyond its designed capacity could lead to overcurrent.
To prevent these problems, the tactile sensor plate includes components such as a PTC Resettable Fuse and a MOSFET. The PTC Resettable Fuse protects the circuit by cutting off the current during overcurrent situations and resetting itself once the issue is resolved. The MOSFET acts as a switch, helping to control the voltage and prevent sudden spikes. These additional components help to protect the tactile sensor plate during both testing and operation.

4.2.3 Silicone Coating

The silicone protects against moisture, dust, and physical damage, extending the tactile sensor plate's lifespan. It also serves as a medium through which displacement is transformed into force. The Hall-effect sensors do not measure force directly, but the distance to a magnetic source. When an external force is applied to the silicone, it compresses, reducing the distance between the magnets and the sensors. This displacement can be converted into force using Hooke's Law, the fundamental principle behind the project's approach.
Ecoflex 00-20 silicone was selected for this purpose. Hardness levels, from 10 to 30 Shore, were tested, and 20 Shore was chosen as the most suitable. Shore hardness indicates how resistant a material is to deformation. Silicone with 20 Shore hardness strikes a balance: it is soft and flexible enough to compress evenly without being too soft, which could reduce sensor protection, or too hard, which could resist pressure and affect measurement accuracy.
One drawback of silicone is that it introduces a hysteresis effect during compression, which can slightly affect the precision of readings. [35]
The sensors and electronic components are soldered directly onto the tactile sensor plate, which is then encapsulated in silicone. This silicone layer protects the components and increases the durability of the device. Magnets are embedded in the silicone layer on the opposite side of the tactile sensor plate from the sensors.
The embedding process was done in two steps. A mold was created to give the silicone its desired shape, including specific slots for the magnets directly above the sensors. First, liquid silicone was poured into the mold, and the tactile sensor plate was submerged into it. The silicone layer on the sensor side is thin, just enough to cover the components. On the opposite side, the silicone is about 2.5 mm thick, with 1 mm deep holes for the magnets. These holes position the magnets 1.5 mm away from the PCB. The total thickness of the tactile sensor plate embedded in the silicone became approximately 8 mm. After the silicone hardened, the magnets were placed into the prepared slots, ensuring they all had the same orientation, to simplify later data processing and modelling. The magnets were embedded on the flat side of the tactile sensor plate because it had no components, allowing the silicone to compress uniformly under force without causing irregularities that could impact the readings. Once all magnets were in place, a second layer of silicone was poured to cover and secure them. To ensure the silicone was uniform and free of air bubbles, it was processed in a vacuum chamber. The result of covering in silicone

is shown on Figure 4.7.

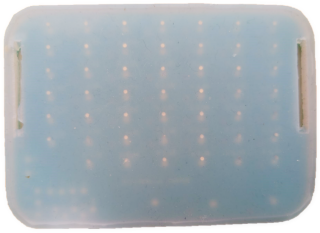


Figure 4.7: The tactile sensor plate covered in silicone.

4.2.4 Raspberry Pi Housing

The Raspberry Pi housing is designed to protect the controller from external factors, such as physical damage, while also allowing it to be securely mounted on the forearm. The housing was designed with the following key factors in mind:

- **Ventilation:** Multiple side and bottom openings for air circulation to prevent overheating.
- **Modular Assembly:** The two-part design allows easy access for maintenance.
- **Minimalistic Surface Design:** The design keeps the surface clean and smooth, without unnecessary shapes or holes, making it easier to clean.
- **Cable Management:** Openings are positioned to allow efficient routing of power and data cables.
- **Mounting Slots:** The case includes dedicated slots on the sides for attaching a strap, allowing mounting around the forearm.

(a) Raspberry Pi housing bottom.

(b) Raspberry Pi housing top.

(c) Raspberry Pi housing bottom.

(d) Raspberry Pi housing top.

Figure 4.8: Raspberry Pi housing.

The design is shown in Figure 4.8. The housing is designed to be 3D-printed using PETG (Polyethylene Terephthalate Glycol). PETG was chosen due to its durability, improved chemical resistance, and smooth surface finish.

4.2.5 Connector and Interface Design

The connection between the tactile sensor plate and the Raspberry Pi is established using the specially designed connector, as shown in Figure 4.6a and 4.6b. This connector is responsible for transferring both power and data signals. The cable length has been chosen as a compromise — as short as possible to minimize noise increase, yet as long as necessary to allow for proper mounting on the hand. The Raspberry Pi is housed in a case designed for this project, allowing for easy mounting and comfortable use. The setup with a tactile sensor plate connected to the Raspberry Pi illustrates Figure4.9

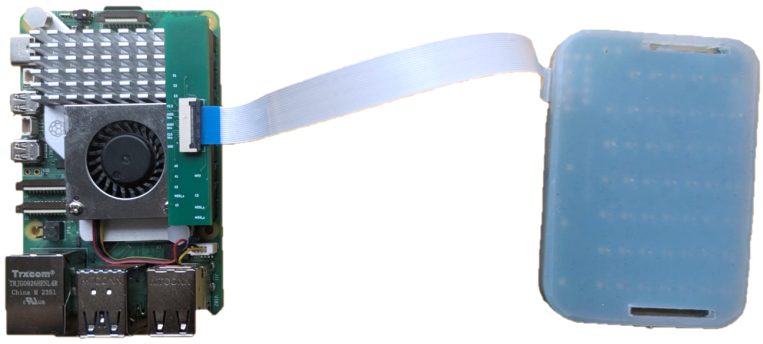


Figure 4.9: tactile sensor plate assembly.

4.2.6 Connection Between Tactile Sensor Plate and Raspberry Pi 5

The hall-effect sensors communicate with the Raspberry Pi 5 using the SPI1 interface. The following GPIO pins are used for communication:

- **MISO:** Used to send data from the sensors to the Raspberry Pi.
- **MOSI:** Used to send data from the Raspberry Pi to the sensors.
- **CLK:** The clock signal synchronises communication between the Raspberry Pi and the sensors.
- **E0, E1, E2:** Enable pins used to control the demultiplexers on the tactile sensor plate, enabling them on low signal.
- **A0, A1, A2, A3:** Address pins select specific sensors via the demultiplexers.
- **3.3V Power:** Supplies power to the tactile sensor plate.
- **Ground:** Provides a common ground on the tactile sensor plate.

The detailed mapping of GPIO pins to functions is shown in Table 4.1.
The detailed mapping of GPIO pins used for IMU communication is shown in Table 4.2. Notice the IMU and MLX90393 share the same power supply lines.

Pin Name	Raspberry Pi GPIO Pin
MISO	GPIO 9
MOSI	GPIO 10
CLK	GPIO 11
E0	GPIO 22
E1	GPIO 23
E2	GPIO 27
A0	GPIO 5
A1	GPIO 6
A2	GPIO 26
A3	GPIO 13
3.3V Power	Pin 1
Ground	Pin 6

Table 4.1: tactile sensor plate connections to Raspberry Pi GPIO pins.

Pin Name	Raspberry Pi GPIO Pin
MISO	GPIO 19
MOSI	GPIO 20
CLK	GPIO 21
CS	GPIO 16
INT1	GPIO 25
INT2	GPIO 12
3.3V Power	Pin 1
Ground	Pin 6

Table 4.2: IMU connections to Raspberry Pi GPIO pins.

4.3 Software System Design

The system is designed as a QT application using C++. The choice of QT was determined by its efficient tools for creating GUI.
Figure 4.10 illustrates the process workflow. On the left side is the tactile sensor plate and the controller - Raspberry Pi 5. Using the QT application, the controller handles communication with the tactile sensor plate. On the right side of the Raspberry Pi are four lines in different blue colours, representing the four threads handling different parts of the system. The small arrows indicate the communication directions between the system components.
The first thread is responsible for initializing the communication between the tactile sensor plate and the Raspberry Pi. This initialization occurs only once, at the beginning of the process. After initialization, the same thread continuously gathers data from the sensors for further processing.
The second thread performs the one-time action of deploying models for all working sensors, giving a total of 45 models, one for each sensor.[2] In addition to model

[2]There are 48 sensors on the tactile sensor plate, but three sensors were faulty and therefore excluded from further modeling and implementation.

deployment, this thread is also responsible for managing the GUI.
The third and fourth threads are dedicated to processing the data collected from the sensors. Each thread processes data from specific sensors by passing the sensor readings through the respective model associated with each sensor.

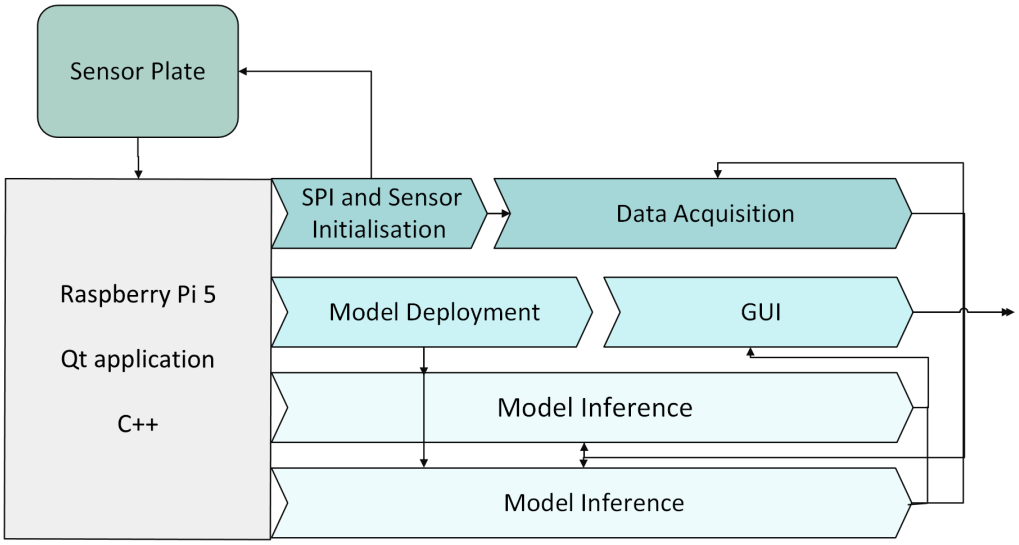


Figure 4.10: System Architecture of the Tactile Sensor Plate System. This diagram illustrates the interaction between key modules in the system, including the flow of data from the sensor plate to the user interface.

Figure 4.11 shows the schematic workflow of the software system, with four modules represented in different colours, starting from the data acquisition module to the real-time visualization module. Each module handles a specific task. This setup makes it easier to develop, test, and update in the future. The sections below

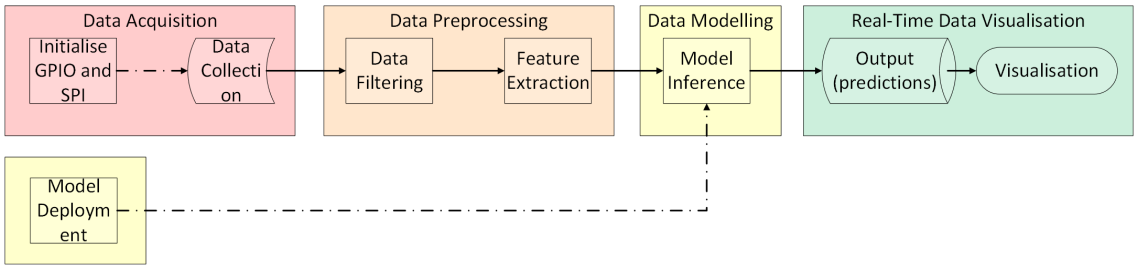


Figure 4.11: System Design: Interaction between software modules

describe each module in detail.

4.3.1 Data Acquisition

The Data Acquisition module is responsible for communication between the sensors and the controller to gather data at the required intervals. It handles the setup of GPIO pins on the Raspberry Pi, initialization of SPI in Mode 3, and configuration of the sensors to operate in burst mode for data collection.

A dedicated C++ library was developed specifically for the Raspberry Pi 5, to establish communication with the sensors and collect data. This library manages the initialization of the SPI interface, activation of burst mode, transmission of commands to the sensors and demultiplexers, and retrieving data at 100 Hz.
Although the IMU has not yet been added to the software, the system design accommodates its inclusion as the project develops.

4.3.2 Data Prerocessing

This module prepares the collected data for further analysis.

Savitzky-Golay Filter

The sensor data is filtered using the Savitzky-Golay filter to reduce noise and smoothing the signal. The result of the applied filter is shown in Figure 4.12. The optimal filter parameters implemented are a window size of 21 and a polynomial degree of 4. These parameters smoothen the noisy data and preserve the fast system's responsiveness.

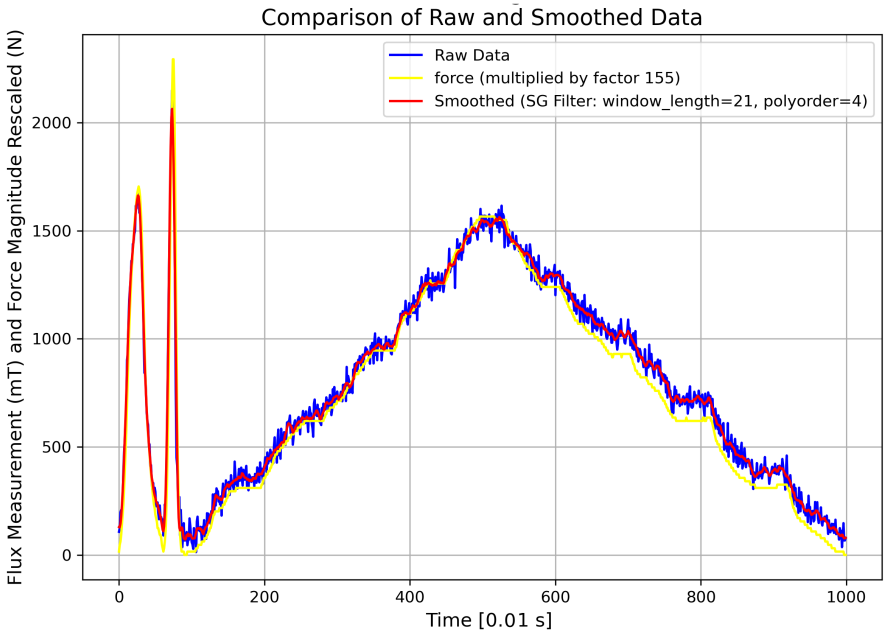


Figure 4.12: Signal Smoothing Using Savitzky-Golay Filter

Offset Calibration

Offset calibration is an important part of the data preprocessing step. It is performed at the beginning of each session, whenever the environment changes or the data shows unusual behaviour. This process adjusts the system considering external magnetic fields and other environmental interferences that could affect the accuracy of the readings.

The calibration works by taking an average of 500 readings for each sensor and each reading type. The mean values are then subtracted from the sensor readings to bring the offsets as close to zero as possible. The calculated offset values are stored in a temporary array, which gives fast access during data acquisition.
The decision to implement offset calibration in this way, rather than writing the offsets to the sensors' volatile memory, was determined by how the tactile sensor plate will be used. Frequent recalibration is necessary since the tactile sensor plate is designed to operate in different environments with varying magnetic fields. Writing to the sensor's memory would require additional steps to modify and store the offsets for each calibration, which may not be as efficient or practical, especially during the developing phase, when frequent recalibration is needed. In this way, the calibration is faster and easier to implement. However, when the device once is a developed product, saving to volatile memory might be considered as well.

4.3.3 Data Modeling

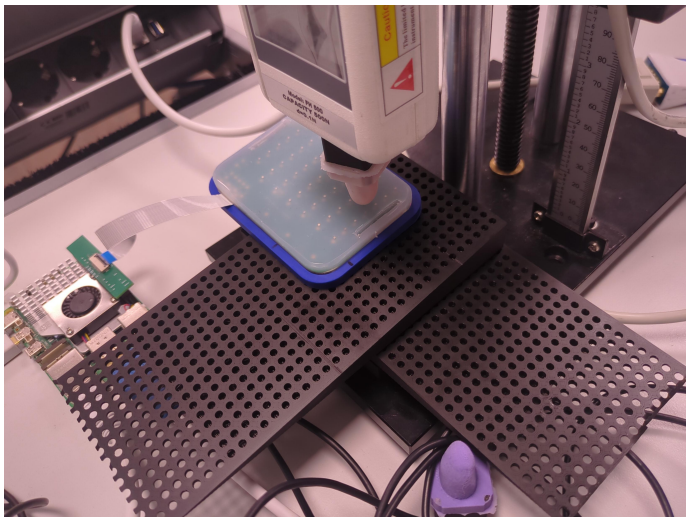
The Data Modeling module is responsible for implementing the designed model to predict the magnitude, position, and direction of applied forces. The initial approach aimed to evaluate whether supervised learning models would be more suitable for force prediction. To evaluate if it is suitable, the decision tree model was first tested on a single sensor, achieving a very good prediction accuracy of approximately 99%. However, while this model performed well in isolated cases, it struggled with handling nonlinear relationships, particularly those arising from the distance between the force application point and the sensor. The model failed to generalize well across different sensor locations because the force does not decrease linearly with distance, t.
Since Random Forest is designed to handle nonlinearity by combining multiple decision trees and averaging their predictions, it was the next model tested. The random forest model demonstrated strong predictive performance, successfully capturing the interactions between force application and sensor readings, even in non-linear cases. Based on these findings, we chose this as our model due to its ability to generalise across different force distributions and maintain high accuracy across all sensors.
Another consideration was to use a CNN. While CNNs have been successfully used for spatial data processing, due to their high computational cost and the need for extensive training data, they became a second choice in case supervised learning models wouldn't be able to simulate the data.
The model uses Random Forest Regression algorithms. The model is built in Python and deployed to C++ using the Open Neural Network Exchange (ONNX) system. ONNX was chosen for its efficiency in managing multiple models in a very short time. ONNX enables integration of the Python model into the C++ environment with minimal overhead and fast inference times (the time it takes to generate predictions once the input data are received). This is important since the project utilises a large number of models.
The system is designed to make it able to, after deploying the models, process sensor data in real-time, predicting the force components and their directions. The output can then be formatted for visualisation, creating graphical representations of the

force components to allow further analysis.

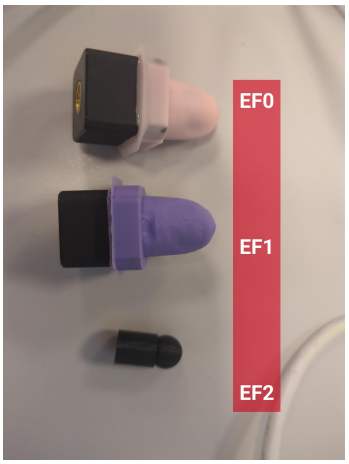
Data Gathering for the Modelling

The process of gathering data, while part of the Data Acquisition module, is described here as it is closely tied to the subsequent modelling steps. The model data was collected by readings from the MLX90393 sensors and synchronizing with force magnitude readings. Synchronization was achieved using timestamps from Qt signals, a simple method that avoids the need for complex synchronization techniques. The data was recorded at a frequency of 100 Hz. Force measurements were taken using the Sauter FH-500 force gauge.

To ensure consistency, a specially designed base was created to enable data collection at equal intervals of 5 x 5 mm and at angles of 5° and 10° in both the x and y directions. Due to the force gauge limitations, the angle interval had to be small and limited to 10°. Figure 4.13a illustrate the data collection procedure. Three different end effectors were used to introduce variability and simulate different finger shapes, as shown in Figure 4.13b.



(a) Data collection process for sensor modeling.



(b) End effectors used for data collection.

Figure 4.13: Data Collection setup.

The force patterns, shown in Figure 4.14, were used to study how the silicone and sensors reacted to different dynamics of the applied force. These patterns were designed to capture different behaviour, such as hysteresis or other possible non-linearities. The pattern below was applied to each force position on the tactile sensor plate.

Forces between 0 and 10 N were applied in different ways, changing how fast, how strong, and how often the force was used. These patterns show peaks and changes that help understand how the sensors react to different force dynamics.

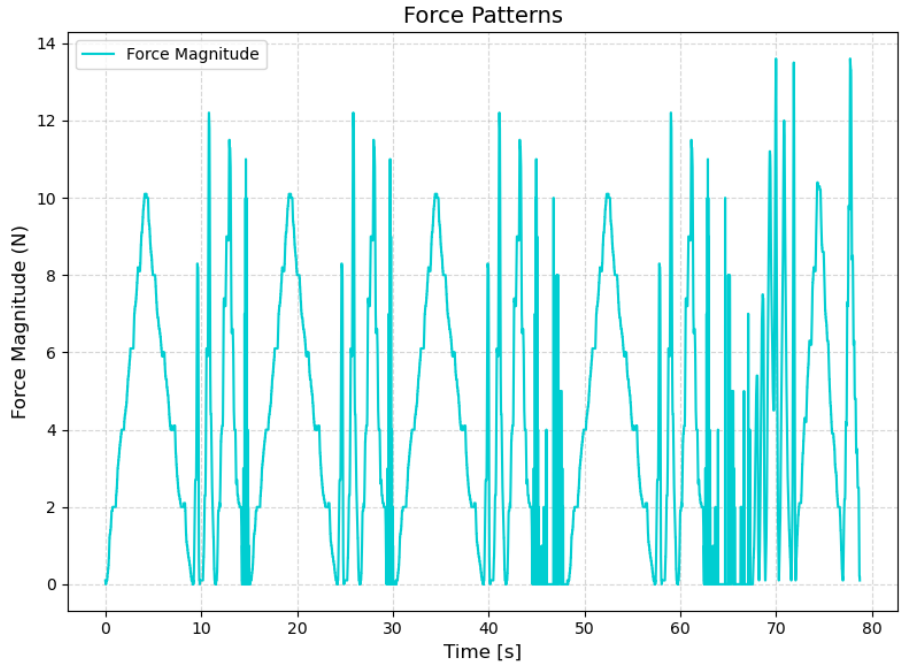


Figure 4.14: Force Patterns during data collection.

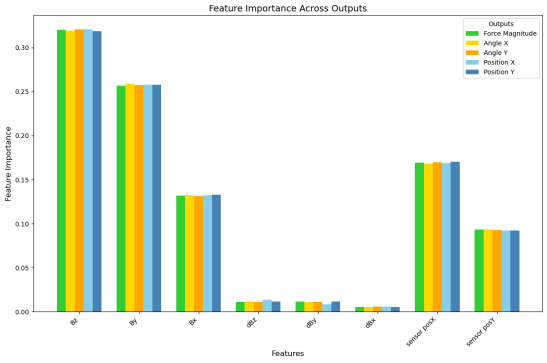
Evaluation of Random Forest Regression Approaches

This section presents the methodology for selecting the final model by analysing the performance of two different approaches: the distributed and combined Random Forest Regression models. The analysis includes modelling techniques, feature evaluation, and performance metrics that led to the choice of the final model.
Two modelling approaches were considered:

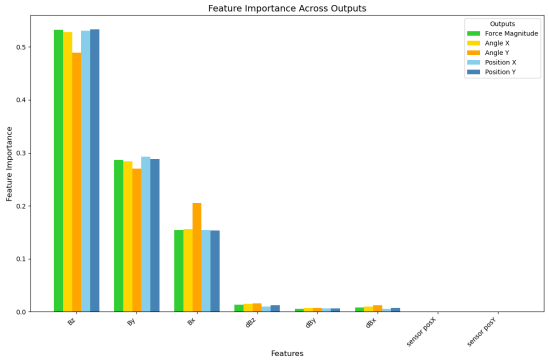
- **Distributed Model Approach:** The modelling task is distributed to multiple smaller models, separate Random Forest Regression models, trained for each sensor on the grid. Each model focuses only on the data collected by its corresponding sensor and is trained to predict force, angle, and position within its region. This specialisation leads to high accuracy and minimises the complexity of data input for each model, as the position of the sensor is not required.
- **Combined Model Approach:** This approach combines all sensor inputs into a single Random Forest Regression model to predict outputs for all sensors on the grid. This approach is built on the full dataset, including the positional data of sensors, to distinguish between readings from different locations. The combined model generalises across the entire grid, making it capable of handling force predictions for all sensors in a unified manner.

In both approaches the input data the analysis started with is a three-dimensional array with dimensions $8 \times 7 \times 7$. The first dimension includes the eight features used by the model: magnetic flux readings in the x, y, and z directions (B_x, B_y, B_z), flux gradients (dB_x, dB_y, dB_z), and the sensor positions on the grid (P_{Sx} and P_{Sy}). The second and third dimensions represent the grid of sensors. The importance of the features is shown in Figure 4.15 and analysis of the modelling approaches is

described below.



(a) Feature importance of the single combined model.



(b) Feature importance of the distributed models.

Figure 4.15: Comparison of feature importance between the single combined and the distributed models.

The feature importance of the combined model shows that B_z remains the most important feature, followed by B_x and B_y, and the contributions of the sensors' positions are significant as well. The importance of the flux gradients is insignificant. Unlike the distributed models, where the importance of the flux features dominates, the combined model utilizes sensor positions as well, which can be explained by the fact that it needs to generalize across the entire sensor grid and distinguish data from different sensors. Distributed models don't need positions because each model works with one specific sensor.

Optimal Model Selection

The performance of the single model is 5,31% worse compared to the distributed approach, as shown in Table 4.3 The poorer performance of the combined model might be because it generalises across all sensors, while the distributed model is specialised in modelling the particular sensor. This generalization results in lower accuracy and higher errors.
The focus is therefore to choose the optimal distributed model, where the performance of models with different feature sets was analysed. The model using all features provides the best accuracy, as expected. However, excluding the sensor positions has minimal impact on performance. The R^2 score for force predictions is only 0.22% lower than the all-feature model, and for position predictions, the R^2 is approximately 1.1% lower. These small differences make the model without positions the best option, especially given its reduced complexity and processing requirements.
The presence of flux gradients was also evaluated. Although the gradients contribute less than 5% to feature importance, they are valuable for accounting for hysteresis effects in the silicone material. Even though the relationship between force and flux is nearly linear for forces up to 10 N, as shown in Figure 4.16, the gradients capture subtle variations that slightly improve position predictions. Excluding the

gradients results in a drop of accuracy of approximately 1.1% for position predictions compared to the model with all features.

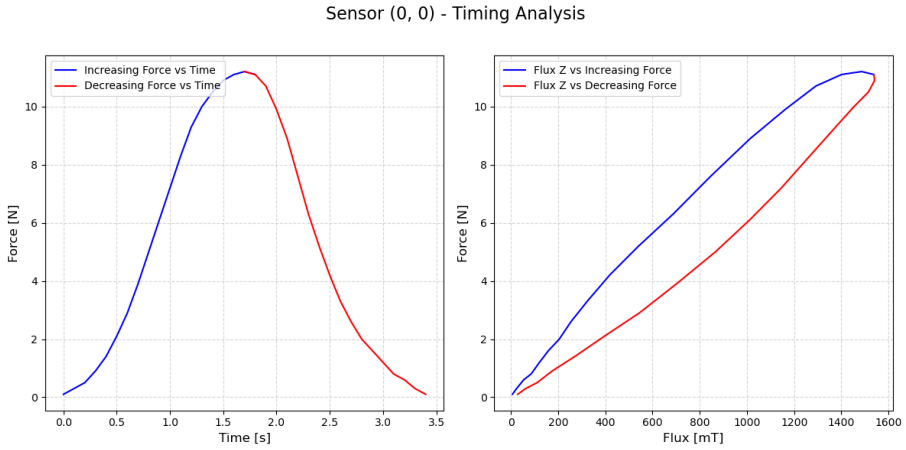


Figure 4.16: Illustration of hysteresis behaviour.

The most optimal model is the distributed model where flux readings are only features. It shows the best balance between accuracy and complexity (which has a direct impact on data processing and inference time), and the slight reduction in accuracy is an acceptable trade-off for faster processing and reduced input size, which is important when the system has to handle so many models at the same time.

4.3.4 Random Forest Model for Force Prediction

Based on the selection of a suitable model for force estimation, the Random Forest Regressor was implemented for each sensor in the 7×7 sensor grid.

Dataset Splitting

The dataset used for training was divided into three subsets:

- **Training Set:** 75%
- **Validation Set:** 20%
- **Test Set:** 5%

Data splitting was performed using `train_test_split` with `random_state = 9` to ensure reproducibility.

Random Forest Model Parameters

The following hyperparameters were used for training each sensor model:

- **Number of Estimators** (*n_estimators*)**:** 40 (number of decision trees)
- **Maximum Depth** (*max_depth*)**:** 20 (limits tree depth to prevent overfitting)
- **Number of Jobs** (*n_jobs*)**:** -1 (utilizes all available CPU cores for parallel training)
- **Random State** (*random_state*)**:** 9

Table 4.3: Comparison of Model Performance Metrics Across Features

Model Feature	Distributed All Features	No Positions	No Gradients	Just Flux	Combined All Features
Force (F)					
MSE	0.0134 ± 0.0209	0.0148 ± 0.0207	0.0148 ± 0.0207	0.0148 ± 0.0207	0.0472 ± 1.0941
MAE	0.0091 ± 0.0070	0.0100 ± 0.0070	0.0100 ± 0.0069	0.0100 ± 0.0069	0.0237 ± 0.2159
R^2 (%)	98.08 ± 2.45	97.86 ± 2.39	97.86 ± 2.39	97.86 ± 2.39	92.73 ± 0.00
Max Error	5.0673 ± 1.7399	5.0752 ± 1.6930	5.0820 ± 1.6859	5.0820 ± 1.6859	14.0116 ± 0.2159
Angle X (Ax)					
MSE	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0008
MAE	0.0003 ± 0.0003	0.0003 ± 0.0002	0.0003 ± 0.0002	0.0003 ± 0.0002	0.0006 ± 0.0064
R^2 (%)	88.38 ± 8.61	88.13 ± 9.21	88.13 ± 9.21	88.13 ± 9.21	64.72 ± 0.00
Max Error	0.1407 ± 0.0589	0.1398 ± 0.0606	0.1398 ± 0.0606	0.1398 ± 0.0606	0.1745 ± 0.0064
Angle Y (Ay)					
MSE	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0005
MAE	0.0002 ± 0.0002	0.0002 ± 0.0001	0.0002 ± 0.0001	0.0002 ± 0.0001	0.0005 ± 0.0049
R^2 (%)	86.05 ± 8.29	86.40 ± 8.55	86.40 ± 8.56	86.40 ± 8.56	59.94 ± 0.00
Max Error	0.1205 ± 0.0528	0.1213 ± 0.0531	0.1214 ± 0.0531	0.1214 ± 0.0531	0.1745 ± 0.0049
Position X (Px)					
MSE	1.8893 ± 2.1809	2.1469 ± 2.4714	2.1482 ± 2.4716	2.1482 ± 2.4716	3.0847 ± 64.3269
MAE	0.1026 ± 0.0876	0.1102 ± 0.0952	0.1102 ± 0.0952	0.1102 ± 0.0952	0.1709 ± 1.7480
R^2 (%)	91.79 ± 4.28	90.69 ± 4.54	90.68 ± 4.53	90.68 ± 4.53	85.37 ± 0.00
Max Error	30.9598 ± 18.3767	30.8169 ± 18.4364	30.8123 ± 18.4306	30.8123 ± 18.4306	62.6332 ± 1.7480
Position Y (Py)					
MSE	0.5970 ± 0.5285	0.6862 ± 0.6167	0.6862 ± 0.6157	0.6862 ± 0.6157	0.9667 ± 21.6722
MAE	0.0551 ± 0.0372	0.0598 ± 0.0411	0.0598 ± 0.0410	0.0598 ± 0.0410	0.0950 ± 0.9786
R^2 (%)	90.10 ± 4.58	89.07 ± 4.78	89.07 ± 4.76	89.07 ± 4.76	86.80 ± 0.00
Max Error	18.7409 ± 11.6715	18.8424 ± 11.6627	18.8415 ± 11.6639	18.8415 ± 11.6639	36.2607 ± 0.9786

Training and Model Storage

Each sensor model was trained separately and saved as a `.pkl` file using `joblib`. To ensure relevant force readings were used for training, a filtering method selected force data within a 6 mm circular radius around the sensor location.
Additionally, the datasets used for training, validation, and testing were stored for each sensor.

Force Prediction and Position Recalculation

The collected force data for the training contains force magnitude and metadata. The metadata is added manually and describes the applied force's positions and angles. Based on this, the model's output includes the applied force's magnitude, angles, and position in the x and y directions. This choice, rather than directly outputting recalculated force in the x, y, and z directions, was made after testing different model configurations. It was found that predicting force with positions and angles in the x and y directions first, and then converting it to the x, y, and z components using the provided angles, achieved better results. This conversion is performed within the Real-Time Data Visualization module.
When a force is applied to the tactile sensor plate, it is not always normal. Instead, the applied force can have an angular offset, which is directed at an angle relative to the plate. The goal is to determine the exact position where the force acts on the tactile sensor plate, given that it is not static but changes its orientation in real-time during measurements.

The force is originally applied in a coordinate system where it acts purely in the normal direction. However, since the tactile sensor plate tilts around the x and y axes, the actual position where the force is measured must be recalculated based on these changing angles.
During real-time measurements, the applied forces change rapidly, and the tactile sensor plate tilts dynamically. This makes it necessary to project the force position from its normal plane onto the tilted surface of the tactile sensor plate, for correct estimation of the applied force's displacement. To map the force position two consecutive transformations are applied:

1. **Roll** (α_x): Rotation around the X-axis (forward and backward tilting).
2. **Pitch** (α_y): Rotation around the Y-axis (left and right tilting).

Since the force is originally applied in a normal direction (F_z), the initial position of the force application in its coordinate system can be represented as:

$$P_f = \begin{bmatrix} P_x \\ P_y \\ 0 \end{bmatrix} \tag{4.3}$$

where P_x, P_y are the untransformed force application coordinates.
The rotation matrix for a rotation α_x around the X-axis is:

$$R_x(\alpha_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_x & -\sin\alpha_x \\ 0 & \sin\alpha_x & \cos\alpha_x \end{bmatrix} \tag{4.4}$$

Applying this to the force position:

$$P'_f = R_x(\alpha_x)P_f = \begin{bmatrix} P_x \\ P_y\cos\alpha_x \\ P_y\sin\alpha_x \end{bmatrix} \tag{4.5}$$

The rotation matrix for a rotation α_y around the Y-axis is:

$$R_y(\alpha_y) = \begin{bmatrix} \cos\alpha_y & 0 & \sin\alpha_y \\ 0 & 1 & 0 \\ -\sin\alpha_y & 0 & \cos\alpha_y \end{bmatrix} \tag{4.6}$$

Applying this to the previously rotated force position:

$$P''_f = R_y(\alpha_y)P'_f = \begin{bmatrix} P_x\cos\alpha_y + P_y\sin\alpha_x\sin\alpha_y \\ P_y\cos\alpha_x \\ -P_x\sin\alpha_y + P_y\sin\alpha_x\cos\alpha_y \end{bmatrix} \tag{4.7}$$

The tactile sensor plate is treated as a two-dimensional surface in this model, the Z-coordinate in the final projected positions is not considered. The force application points on the tactile sensor plate are given by:

$$P'_{Fx} = P_x\cos\alpha_y + P_y\sin\alpha_x\sin\alpha_y \tag{4.8}$$

$$P'_{Fy} = P_y\cos\alpha_x \tag{4.9}$$

In these formulas:

- P_{Fx}, P_{Fy}, P_{Fz} - Original force application positions in the reference frame
- P'_{Fx}, P'_{Fy} - Recalculated force positions in the tactile sensor plate frame
- α_x - Roll angle (rotation around the X-axis)
- α_y - Pitch angle (rotation around the Y-axis)
- R_x - Rotation matrix for roll
- R_y - Rotation matrix for pitch
- R - Total transformation matrix from the force frame to the tactile sensor plate frame

The tactile sensor plate itself is represented as a 2D plane, meaning that the Z-position of the force application is not directly relevant to this projection. The Hall-effect sensors measure flux in the X, Y, and Z directions, but the force positions on the tactile sensor plate are only calculated in X and Y. This assumption holds because the tactile sensor plate is flat, and the forces are measured based on their effect on this surface.

However, if future studies aim to measure force magnitudes from different angles or consider force impact depth (for example, working with IMU implementation), a three-dimensional model including Z-position calculations may be needed.

To variate force data, three different end-effectors shown in Figure4.13b, labelled 0, 1, and 2, were used. Each end effector has a slightly different shape, which affects the offset. First, the offsets were measured during the measurements. When the model was created, these offsets were adjusted, which improved the position accuracy. Figure 4.17 illustrates the layout of the force and sensor positions on the plate after rotation and translation (i.e. angles respective offset application).

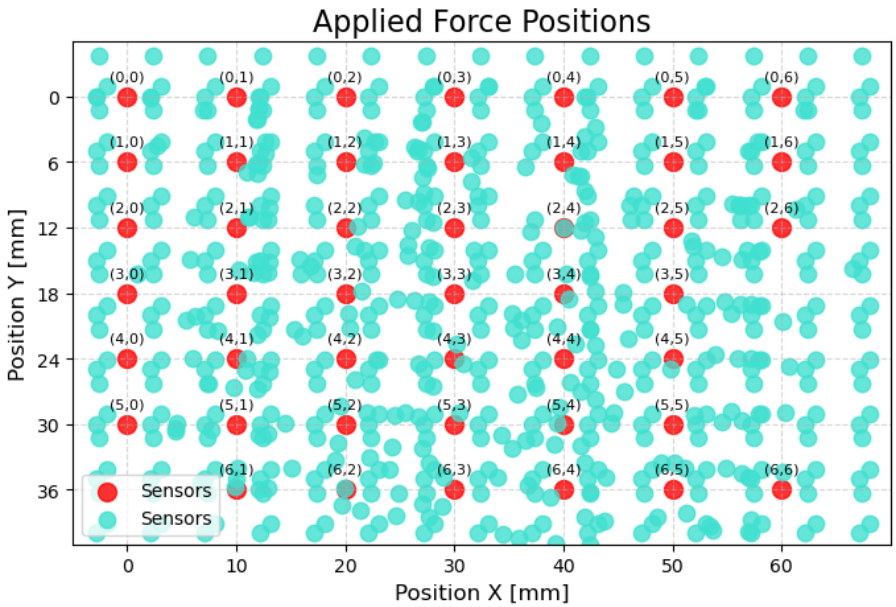


Figure 4.17: Force and sensor positions on the tactile sensor plate. Red points indicate the sensor positions, while blue points represent the applied force positions during data collection for training.

4.3.5 Real-Time Data Visualisation

The Real-Time Data Visualization module converts the output from the Random Forest model into graphical feedback displayed on the GUI. The processed sensor data is shown in real time, giving the user an intuitive visualization of the applied force.

Graphical User Interface (GUI)

The graphical user interface (GUI) is designed to provide an efficient and user-friendly way to interact with the tactile sensor plate system. The GUI consists of a Main Window that presents two primary options: the Developer's GUI and the User's GUI, catering to different user requirements.

Main Window: Launching the application, the main window displays two buttons:

- Developer's GUI: Provides tools for analysis and calibration of the tactile sensor plate.
- User's GUI: Offers a simplified interface focused on visualizing the key outputs of the system and the calibration of the tactile sensor plate.

Developer's GUI: The Developer's GUI is created to allow system developers to test different setups of the tactile sensor plate and its parameters. It includes the following components:

- Sensor Visualization: A graphical representation of the tactile sensor plate, illustrating the flux distribution across all sensors.
- Flux Readings in Text Form: Numerical representation of flux values for each sensor, for precise monitoring and analysis.
- Calibration Button: A function allowing calibration of the sensors, for accurate data acquisition, is important when the environment of the tactile sensor plate changes.
- Force Visualization: A key part of GUI that visualises how forces are applied to the tactile sensor plate, the forces are displayed as x, y, and z components. This visualization is also included in the User's GUI.

User's GUI

The User's GUI is a simplified version of the developers' GUI, aimed at end-users. It focuses on force visualization, offering a graphical representation of the applied forces on the tactile sensor plate. It also has a Calibration Button allowing the user to calibrate the tactile sensor plate.

5

Results

This chapter presents the results of the project, including the sensor plate and software performance. We look at the results of the predicted force parameters and compares them to the applied force parameters. We also look at the results from each step, starting from the preparation of the raw data outcomes to the Graphical User Interface implementation.

5.1 Models Evaluation for Force applied at 0° in both angles

To evaluate the accuracy of the models, an additional set of measurements was collected. The input data was prepared in the same way as the training data, while the output data remained unchanged. The force was applied using the end effector one to five points on the sensor plate. Figure 5.1 illustrates the positions of the applied force. The centre of each applied force is marked as a small blue dot, while the turquoise circle (with a diameter of approximately 9mm) represents the range of the force. Analysing the sensor response and predicted force, at the edges of the turquoise circle, the force magnitude is around 80% of the maximum, rapidly decreasing after this range.

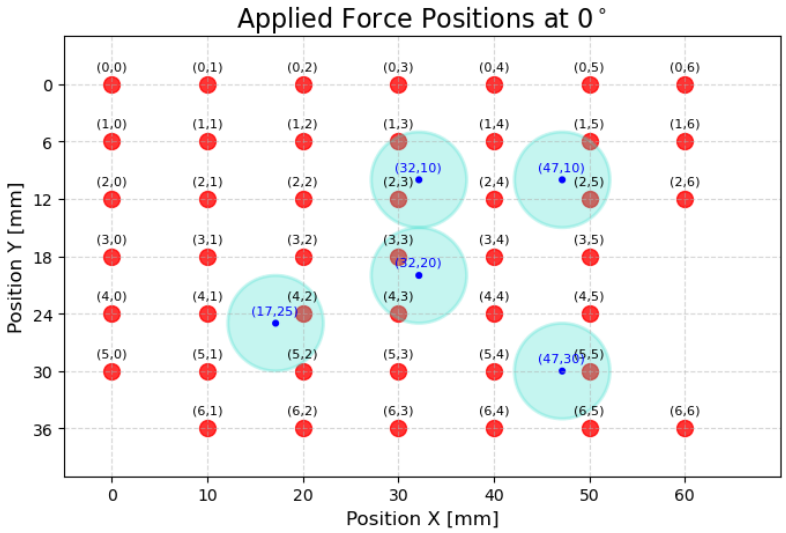


Figure 5.1: Range and position of the applied forces. Force applied at 0° in both angles.

Figure A.16 illustrates how the model estimates the applied force for each sensor. Each subplot corresponds to a specific sensor and displays the force detected within a 6 mm radius and the force predicted by the model. Since multiple sensors might capture the effects of a single applied force because of the radius of 6 mm, the real force readings appear in more than one subplot. Sensors closest to the force application point show a strong alignment between real and predicted forces, while those farther away, but still within the detection radius, register a lower magnitude, accurately reflecting the natural decrease in force distribution over distance.

It is important to note that Figure A.6 does not indicate the exact distance between the applied force and the sensor. Instead, it shows which sensor detected and predicted a force magnitude. If the force was placed near the centre of a sensor, the model predicts the same magnitude. However, when the force was applied further from the sensor's centre, the predicted force magnitude was lower, which is expected, since the largest magnitude should be at the exact point of the force application. The actual coordinates of the force are represented in the position plots, Figure A.7 - A.8, while this plot only indicates which sensor registered the force as the closest. This can be observed in the position plots, where sensors (1,5) and (2,5) not only predict the same force, but also identical x and y coordinates. This is a consequence of the fact that more than one sensor detects the applied force. However, the predicted magnitude might be lower for the sensors, which are further from the centre of the application point. These small variations do not indicate an error in the model but reflect how the force distribution is interpreted. Since each sensor operates independently, the results represent how force is detected by the individual sensor. The behaviour is correct but must be taken into account while designing the graphical interpretation of the predicted force.

The alignment of the estimated and real force positions with the sensor positions is shown in Figures A.7 and A.8, and predicted angles in X and Z directions are shown in Figures A.9 and A.10. The figure shows that the estimated positions and angles match the real ones pretty well.

These observations validate the model's ability to interpret the distribution and intensity of forces applied on the sensor plate, showing the spatial distribution of the readings across the sensor grid and estimating the applied force position with very good accuracy. The results, when compared to the outcomes from the testing data collected in parallel during training (see Figures in A.1 - A.5), show consistent performance, especially considering how force distribution should be interpreted. This means that the models are not overfitted and have very good accuracy for force prediction, and good accuracy for position prediction. While there is room for improvement in predicting the angles and positions, the results are still within an acceptable range when considering the entire sensor plate. This demonstrates that the models capture the overall force distribution well, even if some slight variations appear in specific areas.

Figures A.11 show predicted force based on readings from specific sensors near the force application points. These sensors are chosen based on the Figure A.6.

- **Sensor (3,3), (4,2), (2.5):** The predicted forces align closely with the true forces, $R^2 \approx 96.5\%$, $R^2 \approx 96.0\%$, respectively $R^2 \approx 97.9\%$. Small discrepancies are observed during peaks, where the predicted values do not fully comply

with true values.

- **Sensor (2,3), (5.5):** Demonstrates excellent alignment between the predicted and true forces, with $R^2 \approx 99.15\%$, and $R^2 \approx 99.52\%$.
- **Sensor (1,5):** Shows the lowest alignment between predicted and true forces for the Y-position, $R^2 \approx 82.6\%$. Larger discrepancies are observed, but the overall pattern indicates that the model captures the general trends well. The reason is that the model predicts the distributed force detected by the respective sensor.

Regarding the estimation of positions and angles in X and Y directions, as shown in Figures A.12 - A.15 all sensors exhibit similar behaviour. While the positions are estimated correctly, there is significant instability in the predicted values, fluctuating between 0 and the predicted values. The angles exhibit the same fluctuating behaviour as the positions, but their predicted values do not always match expectations. When no force is applied, the angles show the correct values. However, as soon as force is detected, the angle magnitudes can reach up to 5° in some areas.

5.2 Models Evaluation for Force applied at 5° or 10°

In this experiment, forces were applied at 5° and 10° in at least one direction, using the same method as for 0°. The application points are illustrated in Figure 5.2.

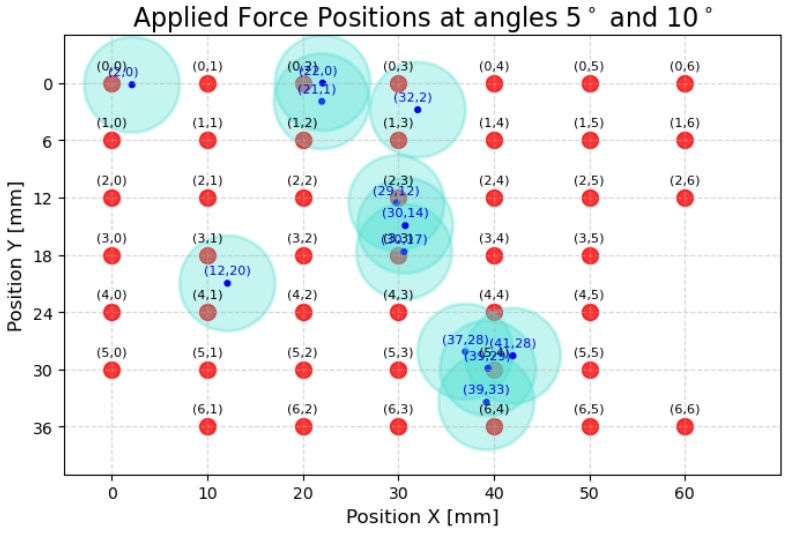


Figure 5.2: Range and position of the applied forces. Force applied at 5° or 10° in at least one of the angles.

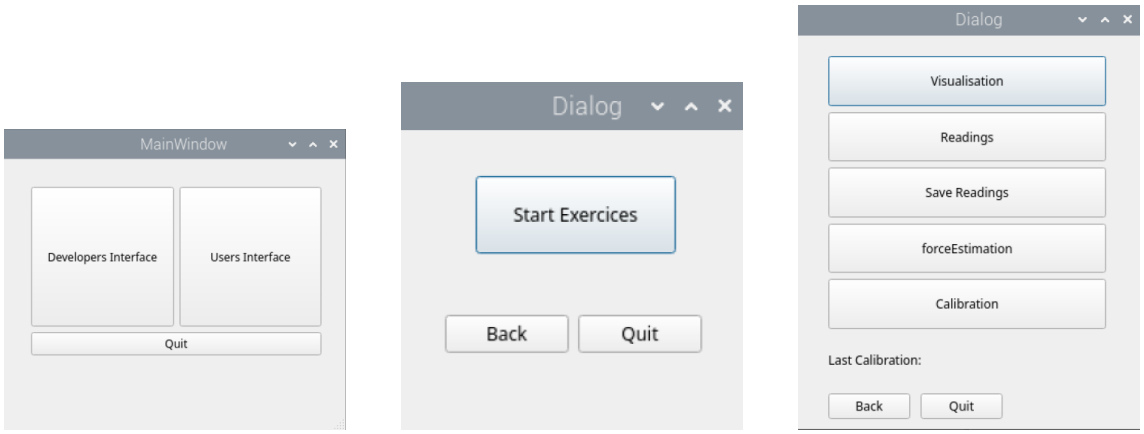
The forces were not decomposed into X, Y, Z components but instead represented as total force magnitudes and angles, which can be converted later. The results from all the sensors are shown in Figures A.16. - A.20. Comparing the results for 5° and 10° with 0°, lower accuracy is observed in some sensors, as depicted in Figure A.21. For example, sensor $(2, 0)$ exhibits an overshoot in predicted force magnitude compared to the real force. However, the general pattern follows the real force well.

One key observation is that the R^2 value is unreliable in some cases. It is visible on Figure A.21b. Specifically, the left force prediction, the magnitude is much lower because the force center was farther from this sensor. This resulted in a lower R^2 value, which does not truly represent the performance. A similar situation occurs in sensors $(0, 3)$, $(2, 3)$, and $(5, 4)$. These sensors were included in the analysis to illustrate how adjacent sensors respond to force.
In these cases, force is detected by more than one sensor since the force radius was slightly larger than half the distance between sensors. Additionally, because X and Y distances are not equal, slight overlapping in the Y direction might have led to force being captured multiple times.
In sensor $(3, 3)$, where no external forces interfere, the R^2 value is 96.44%, which is nearly as high as for 0° forces.
The position results, illustrated in Figures A.22, A.23 reveal an interesting, but unwanted behaviour. Sensor $(0, 0)$ predicts the correct absolute magnitude but in the opposite direction, meaning the position is flipped symmetrically. In general, the position predictions are unstable. While the correct magnitude is often reached, the predictions fluctuate more. Some values are slightly overestimated or underestimated, with errors of up to approximately 5%. This behaviour is visible in sensor $(4, 4)$ (right plot).
A further evaluation of the position predictions will be necessary once visualization tools are available. Some adjustments to the model or shifting mechanisms may be required to improve accuracy. However, at this stage, further corrections are difficult to implement.
The angle predictions show a lower level of accuracy. Sensor $(0, 0)$ fails to detect one of the angles, while other angles are overestimated or underestimated. For example, in plots $(4, 4)$ and $(6, 4)$, the model predicts 10° instead of 5°. This could be due to a miscalculation in the model, but another possibility is an error in data compilation, where an angle of 5° was incorrectly recorded as 10°. Given the equal amplitude in the plots, this explanation appears plausible. Nevertheless, the angle predictions require further improvement to enhance reliability and accuracy. The force predictions have high accuracy for both normal and shear forces, closely following the real values. Position predictions show slightly lower accuracy, with minor deviations of up to 5%, but still provide reasonable estimates. Angle predictions exhibit the largest discrepancies, with some values being overestimated or missing. However, despite these limitations, the overall performance is acceptable.

5.3 Graphical User Interface

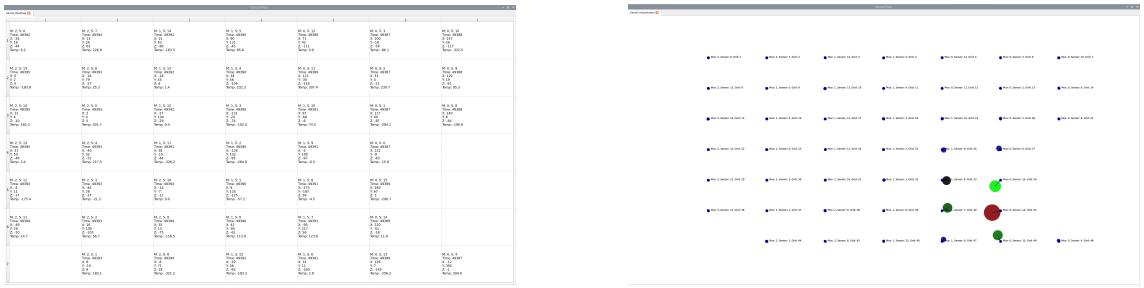
The Graphical User Interface (GUI), opens with the main window (Figure 5.3a), which gives three choices, open Developers GUI, open Users Interface, or Quit the application. After choosing the option for developers, the other window opens, with the menu of buttons as shown in Figure 5.3c. From here the user can choose to visualise the sensor readings by pressing visualisation button (Figure 5.4b), read the sensor data and display them on another window (Figure 5.4a), save the sensor and force readings to a file, estimate the Force, (the functionality of it is not implemented yet) or to perform the offset calibration. The second choice in the main window will

open the Users Window, and the new window opens (Figure 5.3b. After pressing the button, the user will see the graphical visualisation of the applied force and be able to calibrate the sensors (not implemented yet).



(a) Main Window. **(b)** Users Window. **(c)** Developers Window.

Figure 5.3: Graphical User Interface of the application, illustrating the functional windows.



(a) Sensor Readings View. **(b)** Visualization View.

Figure 5.4: The Sensor Readings View and Visualization present data acquisition and graphical representation, respectively.

The Visualization Window, gives a graphical representation of sensor activity during force application, showing how the magnetic flux changes.
The small dark blue circles represent the initial positions of the sensors when no force is applied. As force is applied, the change of the position of the circles represents displacement in the x and y directions, visualise the respective direction of the applied force.
The size and colour of the circles indicate the strength of the magnetic flux, and by this the applied force:

- **Dark blue**: No force.
- **Green**: Moderate force.
- **Red**: Maximum force (up to 10 N).

The real-time graphical representation of magnetic flux sensor readings while applying pressure to a sensor plate is illustrated in Figure (5.5). The setup consists of

a wearable microcontroller unit connected to the tactile sensor plate, transmitting data to a GUI for visualization. The colour variations in the GUI indicate changes in magnetic flux distribution, corresponding to the applied force.

Figure 5.5: Real-Time Magnetic Flux Visualization During Tactile Sensor Plate Compression

6

Discussion

This chapter discusses and analyses the outcomes of the project. It describes the challenges experienced during the development and how they were solved. Topics include hardware construction, software development, and integration issues. It also provides insights into lessons learned during the process and suggestions for future improvement.

Sensor Plate Hardware Evaluation and Suggested Improvements

The sensor plate performed well from both electrical and hardware perspectives. The Hall-effect sensors functioned correctly, and even the IMU, which was not fully implemented in the current functionality, operated as expected. Debugging points added to the PCB design were very helpful during development, allowing easier testing and enabling reliable signal readings from the sensors.

However, during the development stage, it became clear that the chosen connector and flat cable were not ideal for this project. The flat cable was highly sensitive to repeated disconnections, quickly becoming damaged if inserted and removed multiple times. Additionally, the cable was prone to bending issues, which further compromised its durability. To address this, it is recommended to replace the current flat cable and connector with a more robust solution that can withstand frequent disconnections and bending.

Another suggested improvement involves separating the Supply Voltage and Ground lines on the connector, which are side by side in the current design. This increases the risk of a short circuit if the flat cable is connected improperly. This issue arose from inexperience during the PCB design process, highlighting the importance of carefully considering connector design for future iterations.

Lastly, one GPIO pin used for a demultiplexer, specifically pin 17, was discovered to be unsuitable. This pin could not toggle in a controlled manner because it is reserved by SPI 0. To resolve this temporarily, the connection was bridged to pin 23. For future versions, it is recommended to reassign this demultiplexer to a different GPIO pin not reserved by other functions.

6.1 Force Predictions

The results of this study show that the model is performing well in predicting force magnitudes and positions across the sensor plate. However, certain observations highlight areas for improvement to increase the system's overall accuracy.

The predicted forces generally align very well with the true forces, as evidenced by the high R^2 values for most sensors. For example, sensors from the testing phase achieved R^2 values on average approximately 98%, demonstrating strong predictive ability. The small discrepancies observed during force peaks suggest that the model struggles slightly to fully capture rapid changes in magnitude. One of the controlled sensors showed poor prediction accuracy. By examining the character of the misprediction, this issue could be due to the greater distance to the force application, and the sensor detected a smaller force amplitude. However, further investigations should be done to eliminate other possible reasons, such as incorrect model estimation or faulty sensors. These discrepancies could be addressed by improved training data representation. Collecting additional training data for different positions, particularly in closer alignment with sensor locations, would help refine force localization. After analysing the predictions versus application points for testing and training forces, it was observed that forces applied closer to the center of the sensor grid typically achieved higher accuracy. This is likely because sensors closer to the application point detect higher flux readings. Conversely, sensors positioned further from the force application point detect lower force magnitudes, which is the expected and correct behaviour. This pattern confirms that the model follows the expected physical response of the sensor grid.

Shear force predictions also performed well, demonstrating a high level of accuracy comparable to normal forces. The estimated shear force magnitudes generally followed the true values, with only minor deviations. However, slight variations were observed in some sensors, suggesting that additional model fine-tuning could enhance predictive performance in these cases.

The other issue is position and angle instability. While the estimated positions are generally accurate, the instability is present, with predictions fluctuating between zero and the correct values. This instability reduces the reliability of the system for precise force localization. Suggested improvements include implementing stabilization algorithms to smooth position predictions and reduce fluctuations. This could be achieved by holding positions steady for a short period, based on the frequency of fluctuations. The angle predictions, however, show lower accuracy and require further analysis and testing. The discrepancy could be even due to limitations in the Force Gauge, which may not be optimized for measuring forces other than normal.

Some of these modifications could not be implemented due to time constraints, but they would likely improve force prediction accuracy across all features.

6.2 Challenges during System Implementation

Hardware Development

The development of the device began with the design of the first version of the sensor plate PCB, which included Hall-effect sensors and an IMU. However, soldering the small components showed up to be overwhelming.
At this stage, no pre-existing code for the MLX90393 sensors was available for the Raspberry Pi 5 in C++, and the functionality of the tactile sensor plate had not been yet tested.
Further complications came from faulty GPIO pins on the Raspberry Pi 5 producing incorrect signals. This made debugging even more challenging during the early stages of development.
That is why the second version of the PCB was ordered with pre-soldered components. This updated version also has several improvements. These included the addition of more debugging points, adjustments to the connector layout, and the incorporation of fuses and a MOSFET for protection.
To isolate the problem, the code was developed for a single sensor using SparkFun MLX90393 Magnetometer Breakout-ready PCB with the MLX90393 sensor. The code worked successfully on the test PCB, but the root cause of the issue with the custom sensor plate design was still not found. Eventually, it was found that the faulty GPIO pins on the Raspberry Pi were the reason for the misleading results.
This process taught the importance of systematic testing and error isolation in hardware development. Testing individual components, verifying connections, and identifying errors step by step led to finding the source of the problem and its solution.

Software Development

The software was developed as a multithreaded C++ application using the Qt framework. It handles data acquisition, preprocessing, modelling, and visualization.
A major challenge during this phase was creating a new library for the MLX90393 sensor to set up and control the sensors. Existing libraries were designed for Arduino platforms, and no library was available for the Raspberry Pi 5.
Another difficulty was that the Raspberry Pi 5 was still a new release. It used a different chipset than previous Raspberry Pi versions, and many libraries from earlier versions were either incompatible or unavailable. This turned out to be a common issue experienced by many Raspberry Pi 5 users. Resolving these challenges took longer than expected, as it required understanding the differences between the Raspberry Pi 5 and the familiar Raspberry Pi 4.
The initial timeline for the project assumed the existence of a compatible library for the Raspberry Pi and did not account for the differences between Raspberry Pi versions. These differences, particularly with the Raspberry Pi 5's new chipset, created unexpected challenges that required additional time for adaptation and development.
Despite these challenges, the integration of the Qt framework contributed to the soft-

ware's usability and visualization possibilities. Qt offers a wide range of intuitive and user-friendly tools for building graphical user interfaces, making the development process more efficient. Implementing the GUI in Qt allowed for real-time data visualization, helping to better understand sensor outputs and system behaviour. The next step involves fully integrating the model into the system and completing the visualization features, ensuring the sensor plate displays and interacts with force application and distribution.

Insights

This project faced many challenges during its development, but these challenges became valuable learning experiences. Through this process, key lessons were learned, such as how to test a product systematically, find the sources of errors, and fix them effectively. Creating a new library for the MLX90393 sensors based on datasheet information was especially valuable, as it required understanding how to write to registers and configure the device from scratch.

The debugging process also improved skills in isolating problems, testing individual components, and interpreting results step by step. The challenges with the Raspberry Pi 5 and the need to adapt to its differences from previous versions provided deeper knowledge about hardware and software compatibility issues. Additionally, the improvements to the PCB design taught the importance of incorporating debugging points and protective components early in the design phase.

Although the issues led to delays, they strengthened problem-solving skills and provided hands-on experience in hardware and software integration. Summarising, the challenges improved both technical and practical knowledge, which will benefit future projects.

6.3 Conclusion

This section answers the research questions named at the beginning of the project, by summarising the key findings and insights gained from the development and testing of the sensor plate.

What accuracy can the sensor plate achieve in measuring the force and direction?

The sensor plate demonstrated strong accuracy in force estimation. The Random Forest models used for force prediction achieved high R^2 values, between approximately 96% and 99% for force magnitude and good alignment in certain position predictions. However, some instability was observed in the angles and position estimations, particularly with fluctuations in the predicted values. Despite these discrepancies, the sensor plate showed the ability to capture the general trends in force magnitude and direction very well.

What resolution is needed to ensure accurate force measurements?

The resolution provided by the Hall-effect sensors in the sensor plate was sufficient for accurate force measurement. The sensor resolution of 10×6 mm detected forces with high sensitivity, and the machine learning models were able to interpret the data effectively. This shows that the dimension of 10 mm is sufficient to detect force impact on the sensor plate in the current constellation. It might be even possible to decrease the resolution by increasing the gain from the sensors, this evaluation though has not been tested yet.

What design will make the sensor plate easy to clean and maintain?

The sensor plate's design incorporates durable materials and a flat surface, making it easy to clean and maintain. Silicone is particularly advantageous, as it is widely utilised in medical applications due to its resistance to bacterial and fungal growth, ease of cleaning, and waterproof properties. However, a notable weak point is the connector, which could be further improved. Adding a silicone plug to protect the connector during exposure to water would enhance its durability and make it safe for washing. However, it is worth noting that the maintaining and cleaning aspect of the Raspberry Pi housing was not a primary focus of this project.

What environmental conditions should the sensor plate handle?

The sensor plate was designed to function reliably under standard indoor conditions. Testing demonstrated that the regularly performed offset-calibration process successfully neglects the effects of changes in the magnetic field, ensuring stable performance. Additionally, the silicone cover provides effective protection against humidity, enhancing the durability of the sensor plate. The temperature exposure has not yet been tested. Even here, the environmental robustness of the Raspberry Pi was not a primary focus of this project.

Bibliography

[1] R. Drake, *Gray's Anatomy for Students*, 3rd. Churchill Livingstone, 2014, p. 1161, 2015 BMA Medical Book Awards Highly Commended in Basic and Clinical Sciences Category., ISBN: 9780702051333.

[2] G. Rahmani, J. Martin-Smith, and P. Sullivan, "The avocado hand," *Irish Medical Journal*, vol. 110, 2018-01.

[3] L. Evertsson, C. Carlsson, C. Turesson, M. S. Ezer, M. Arner, and C. M. Navarro, "Incidence, demographics and rehabilitation after digital nerve injury: A population-based study of 1004 adult patients in sweden," *PLOS ONE*, vol. 18, no. 4, pp. 1–17, 2023. DOI: 10.1371/journal.pone.0283907. [Online]. Available: https://doi.org/10.1371/journal.pone.0283907.

[4] T. P. Tomo, S. Somlor, A. Schmitz, S. Hashimoto, S. Sugano, and L. Jamone, "Development of a hall-effect based skin sensor," *Waseda University Tokyo, Japan, Instituto de Sistemas e Robótica, Instituto Superior Técnico Lisbon, Portugal*, 2015.

[5] W. Tao, T. Liu, R. Zheng, and H. Feng, "Gait analysis using wearable sensors," *Sensors (Basel, Switzerland)*, vol. 12, pp. 2255–83, 2012-12. DOI: 10.3390/s120202255.

[6] D. Kubo, Y. Yokokohji, and S. Okamoto, "Increasing sensitivity of magnetic tactile sensors by optimizing arrangement of pm array," *IEEE Robotics and Automation Letters*, vol. 7, pp. 7348–7354, 3 2022, Accessed on 2024-12-14. DOI: 10.1109/LRA.2022.3169883.

[7] United Nations, *Sustainable Development Goals*, Accessed: 2025-02-08, 2025. [Online]. Available: https://sdgs.un.org/goals.

[8] Department of Energy and Technology, *What is life cycle assessment?* https://www.slu.se/en/departments/energy-technology/research/lca/, Accessed: Feb. 12, 2024, Swedish University of Agricultural Sciences (SLU), 2022.

[9] P. Yu, W. Liu, C. Gu, X. Cheng, and X. Fu, "Flexible piezoelectric tactile sensor array for dynamic three-axis force measurement," *Sensors*, vol. 16, no. 6, p. 819, 2016. DOI: 10.3390/s16060819. [Online]. Available: https://www.mdpi.com/1424-8220/16/6/819.

[10] H.-K. Lee, J. Chung, S.-I. Chang, and E. Yoon, "Normal and shear force measurement using a flexible polymer tactile sensor with embedded multiple capacitors," *Journal of Microelectromechanical Systems*, vol. 17, no. 4, pp. 934–942, 2008. DOI: 10.1109/JMEMS.2008.923615.

[11] T. Li, H. Luo, L. Qin, *et al.*, "Flexible capacitive tactile sensor based on micropatterned dielectric layer," *Small*, vol. 12, no. 36, pp. 5042–5048, 2016. DOI: 10.1002/smll.201600742.

[12] B. Ward-Cherrier, N. Pestell, L. Cramphorn, *et al.*, "The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies," *Soft Robotics*, vol. 5, no. 2, pp. 220–232, 2018. DOI: 10.1089/soro.2017.0052.

[13] Q. Gui, L. Tao, and H. Liu, "Highly interpretable representation for multidimensional tactile perception," *npj Flexible Electronics*, vol. 7, pp. 1–10, 2024, Accessed on 2024-12-14. DOI: 10.1038/s41528-024-00325-z.

[14] H. Kristanto, P. Sathe, A. Schmitz, T. P. Tomo, S. Somlor, and S. Sugano, "A wearable three-axis tactile sensor for human fingertips," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4313–4320, 2018. DOI: 10.1109/LRA.2018.2858163.

[15] X. Li, R. Deng, W. Jiao, *et al.*, "A high-sensitivity magnetic tactile sensor with a structure-optimized hall sensor and a flexible magnetic film," *IEEE Sensors Journal*, vol. 24, no. 10, pp. 15 935–15 943, 2024. DOI: 10.1109/JSEN.2024.3391786.

[16] J. Farah, N. Baddour, and E. Lemaire, "Design, development, and evaluation of a local sensor-based gait phase recognition system using a logistic model decision tree for orthosis-control," *Journal of NeuroEngineering and Rehabilitation*, vol. 16, no. 1, p. 22, 2019. DOI: 10.1186/s12984-019-0486-z. [Online]. Available: https://doi.org/10.1186/s12984-019-0486-z.

[17] S. Ranasingh, T. Pradhan, and K. R. Dhenuvakonda, "Calibration and frequency estimation in sensors for electrical parameter measurement using regression and metaheuristic based models," *Expert Systems*, vol. 40, no. 3, e13208, 2023. DOI: https://doi.org/10.1111/exsy.13208. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/exsy.13208. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.13208.

[18] H. A. Radi and J. O. Rasmussen, "Magnetic fields," in *Principles of Physics: For Scientists and Engineers*. Cham: Springer Nature Switzerland, 2024, pp. 773–797, ISBN: 978-3-030-48028-8. DOI: 10.1007/978-3-030-48028-8_25. [Online]. Available: https://doi.org/10.1007/978-3-030-48028-8_25.

[19] J. Walker, D. Halliday, and R. Resnick, *Principles of Physics*, Twelfth edition, extended. Hoboken, NJ: Wiley, 2023, ISBN: 9781119820611.

[20] B. J. Goodno and J. M. Gere, *Mechanics of Materials*, Enhanced ninth edition, SI edition. Boston, MA: Cengage, 2021, ISBN: 9780357377857.

[21] T. V. Geetha and S. Sendhilkumar, *Machine Learning: Concepts, Techniques and Applications*, First edition. Boca Raton, FL: Chapman & Hall/CRC Press, 2023, eBook available online. Accessed 13-Dec-2024., ISBN: 9781003290100. [Online]. Available: https://doi.org/10.1201/9781003290100.

[22] S. Brunton, *Physics informed machine learning: High level overview of ai and ml in science and engineering*, Accessed: 13-Dec-2024, YouTube video, 2024-02. [Online]. Available: https://www.youtube.com/watch?v=JoFW2uSd3Uo.

[23] R. W. Schafer, "What is a savitzky-golay filter? [lecture notes]," *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 111–117, 2011. DOI: 10.1109/MSP.2011.941097.

[24] L. Smith, *A tutorial on principal components analysis*, Cornell University, Available at: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf, 1997.

[25] M. Usach, "Spi interface application note (an-1248)," Analog Devices, Tech. Rep., 2015, Rev. A. [Online]. Available: https://www.analog.com/en/technical-documentation/application-notes/an-1248.html.

[26] B. Paradigm, "Using spi protocol at 100 mhz: Understanding the constraints of using spi protocol at higher frequencies," Byte Paradigm Technical Note, Tech. Rep., 2024, Accessed: 2024-12-14. [Online]. Available: http://www.byteparadigm.com/products/spi-storm/.

[27] N. Semiconductors, "I2c-bus specification and user manual," NXP Semiconductors, Tech. Rep., 2021, Rev. 7, October. [Online]. Available: https://www.nxp.com/docs/en/user-guide/UM10204.pdf.

[28] R. P. Ltd, "Raspberry pi 5 product brief," Raspberry Pi Ltd, Tech. Rep., 2024-08, Features a 64-bit quad-core Arm Cortex-A76 processor running at 2.4GHz. [Online]. Available: https://pip.raspberrypi.com.

[29] S. Electronics, *Raspberry pi 5 - sparkfun product image*, Licensed under CC BY 2.0. https://creativecommons.org/licenses/by/2.0/, 2024. [Online]. Available: https://www.sparkfun.com/products/23551.

[30] M. T. NV, *Mlx90393 triaxis® magnetic node datasheet*, version Rev. 010, Accessed on 2024-12-14, 2023. [Online]. Available: https://www.melexis.com/en/documents/documentation/datasheets/mlx90393-datasheet.

[31] STMicroelectronics, *Lsm6dso16 inertial measurement unit (imu) datasheet*, 2024. [Online]. Available: https://www.st.com/resource/en/datasheet/lsm6dso16.pdf.

[32] STMicroelectronics, *Inemo inertial module: Always-on 3d accelerometer and 3d gyroscope*, 2019. [Online]. Available: https://www.mouser.se/datasheet/2/389/lsm6dso-1393615.pdf.

[33] Nexperia, *74hc/hct154 4-to-16 line decoder/demultiplexer datasheet*, 2024. [Online]. Available: https://assets.nexperia.com/documents/data-sheet/74HC_HCT154.pdf.

[34] P. Ghosh, *Applying ipc-2221 standards in circuit board design*, Published on Sierra Circuits' ProtoExpress Blog, Accessed: 2024-04-30, 2023. [Online]. Available: https://www.protoexpress.com/blog/ipc-2221-circuit-board-design/.

[35] I. Smooth-On, *Ecoflex series technical datasheet*, Revision Date: 13-Sep-2024, 2024. [Online]. Available: https://www.smooth-on.com/products/ecoflex-series/.

A

Appendix 1

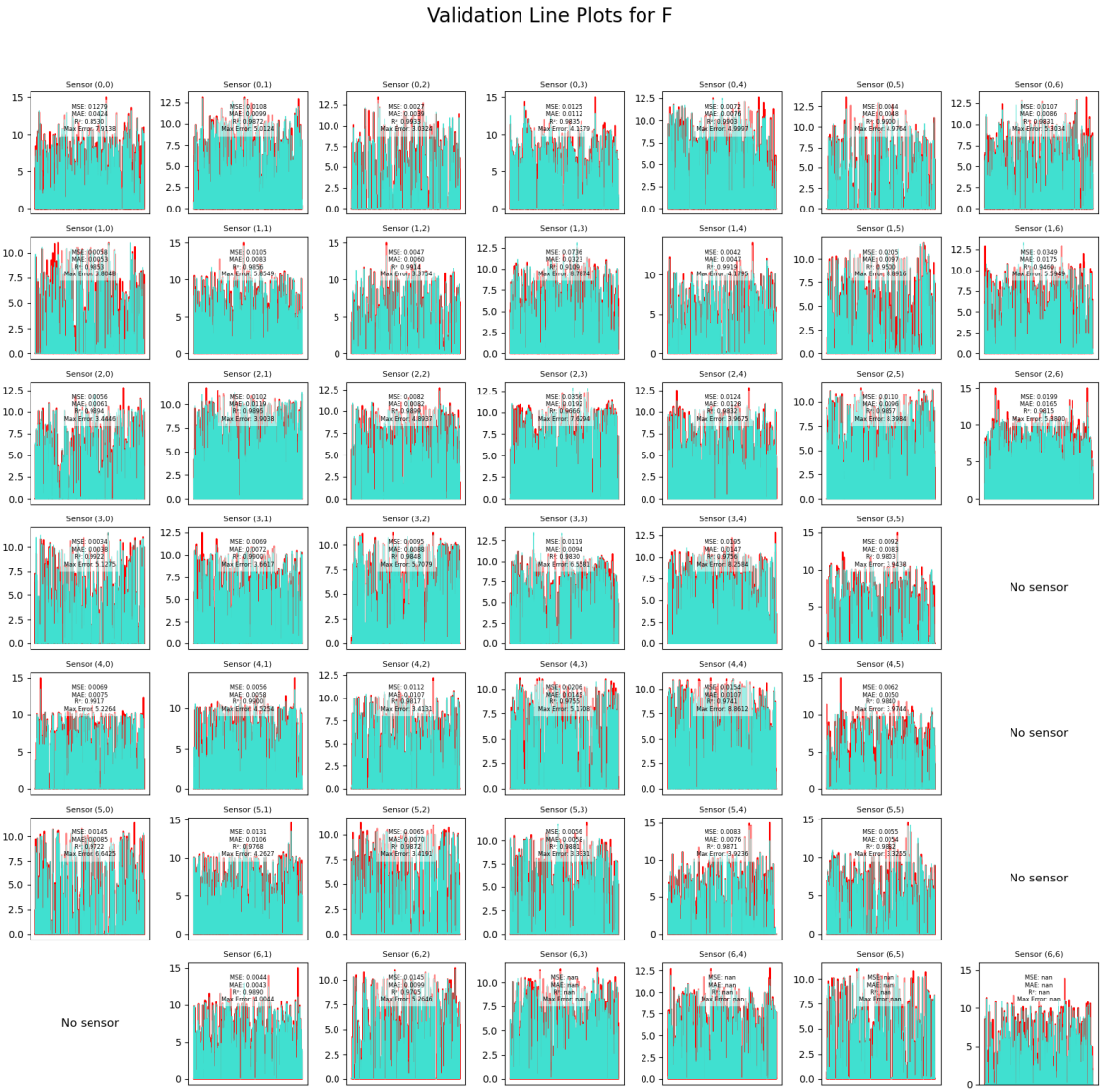


Figure A.1: Force magnitude: The red colour represents the real force, the turquoise colour represents the predicted force values. Data for testing collected simultaneously with the training data.

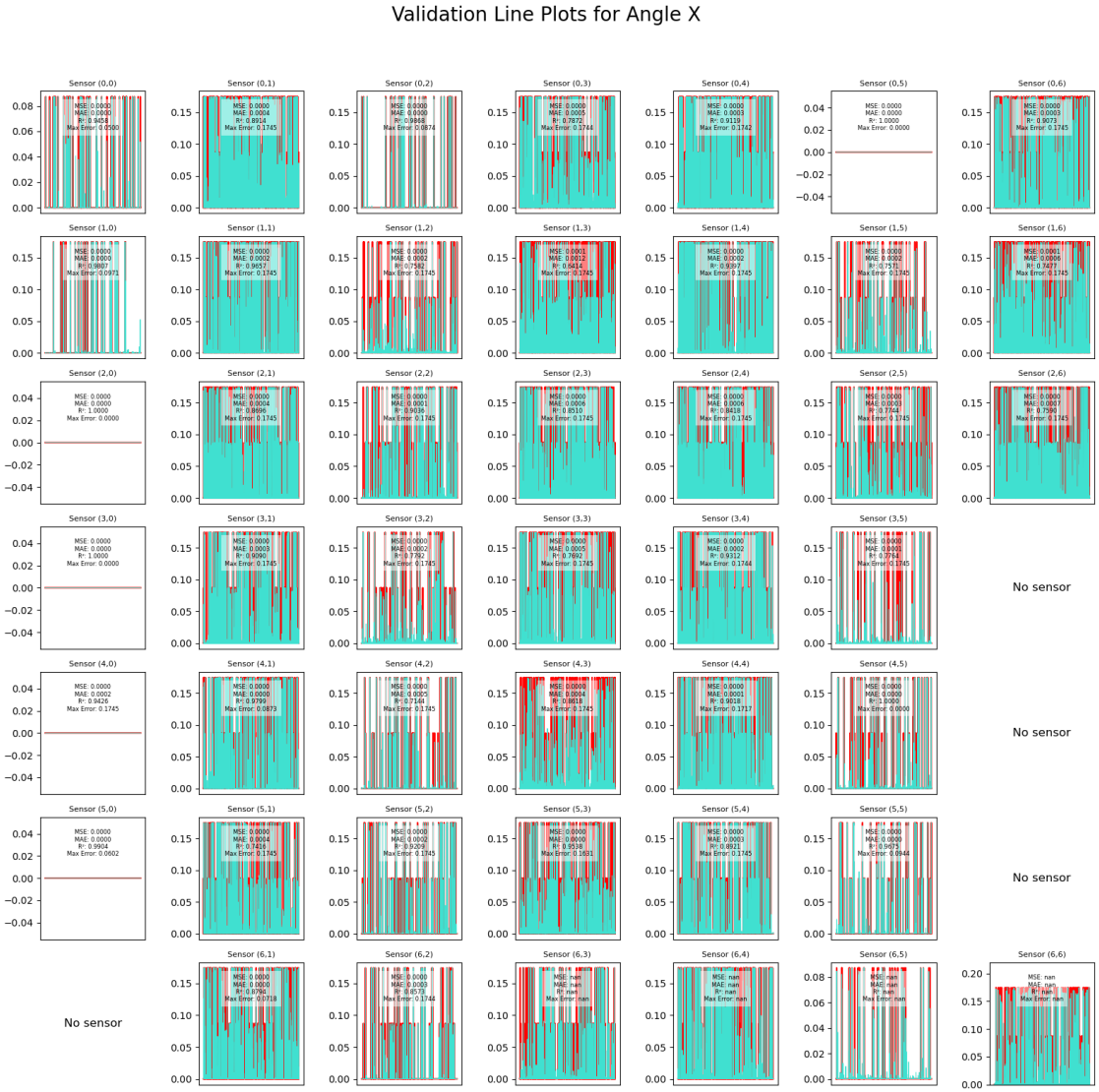


Figure A.2: Angle X: The red colour represents the real angles, the turquoise colour represents the predicted angles values. Data for testing collected simultaneously with the training data.

Figure A.3: Angle Y: The red colour represents the real angles, the turquoise colour represents the predicted angles values. Data for testing collected simultaneously with the training data.

Figure A.4: Position X: The red colour represents the real position, the turquoise colour represents the predicted positions values. Data for testing collected simultaneously with the training data.

Figure A.5: Position Y: The red colour represents the real position, the turquoise colour represents the predicted positions values. Data for testing collected simultaneously with the training data.

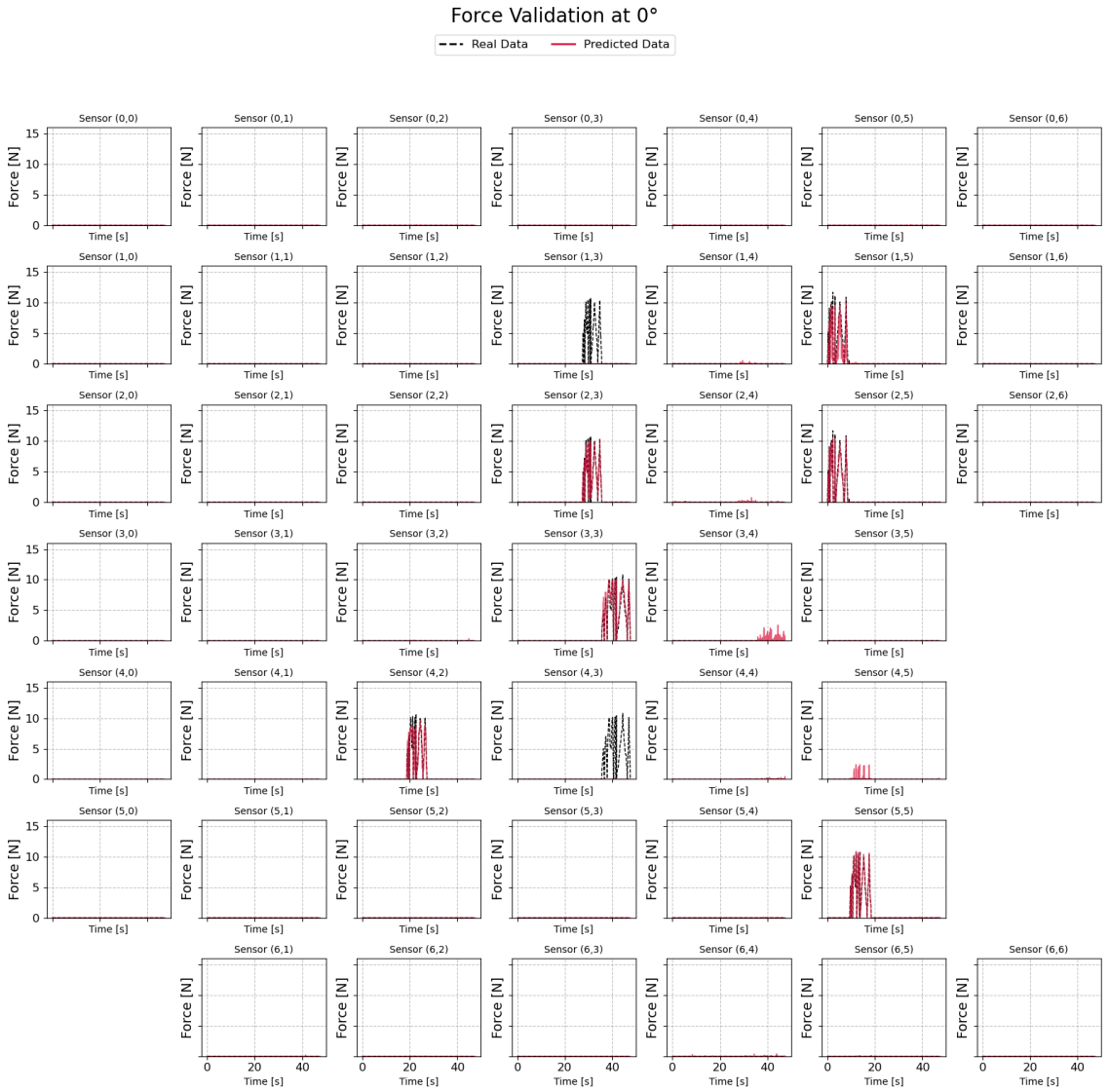


Figure A.6: Comparison of predicted F from each sensor's respective model with actual force readings captured within a 6 mm radius. The figure illustrates how each sensor estimates the applied force, showing both direct readings at the force application point and the distributed predicted force based on readings from surrounding sensors. Force applied at 0° in both angles.

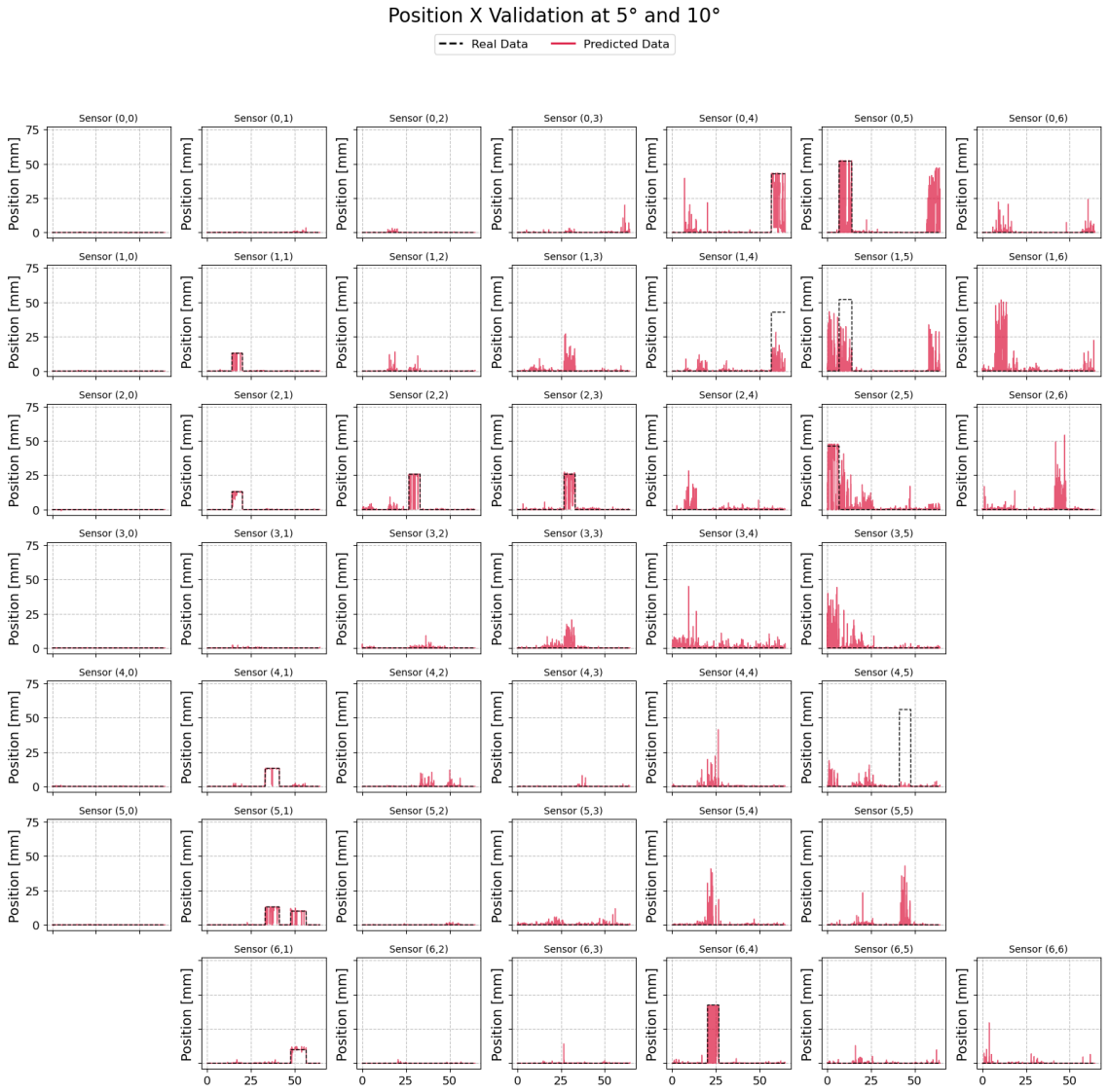


Figure A.7: Estimated and True Force Positions in X across the entire sensor plate, based on readings from each sensor. Force applied at 0° in both angles.

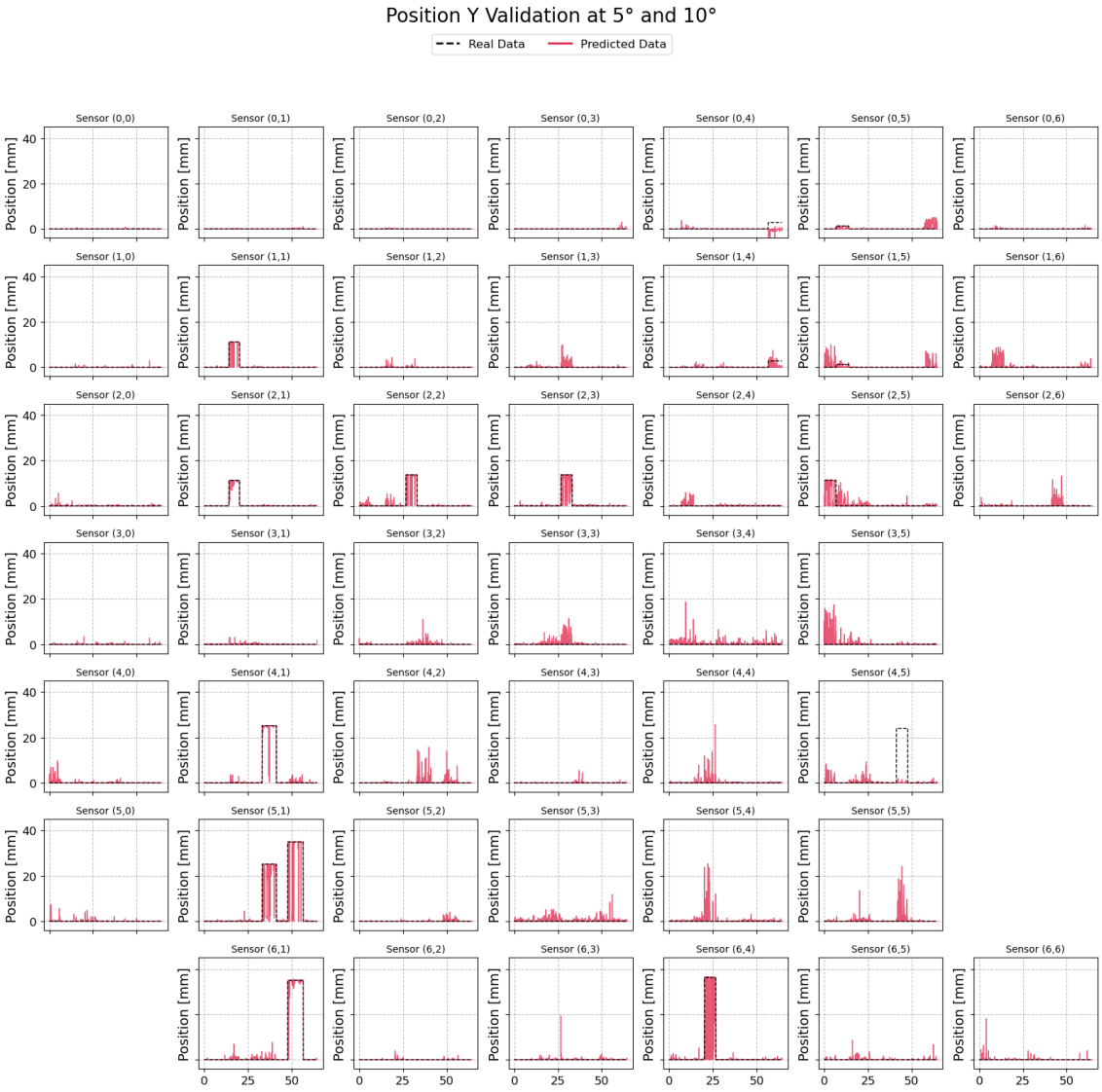


Figure A.8: Estimated and True Force Positions in Y across the entire sensor plate, based on readings from each sensor. Force applied at 0° in both angles.

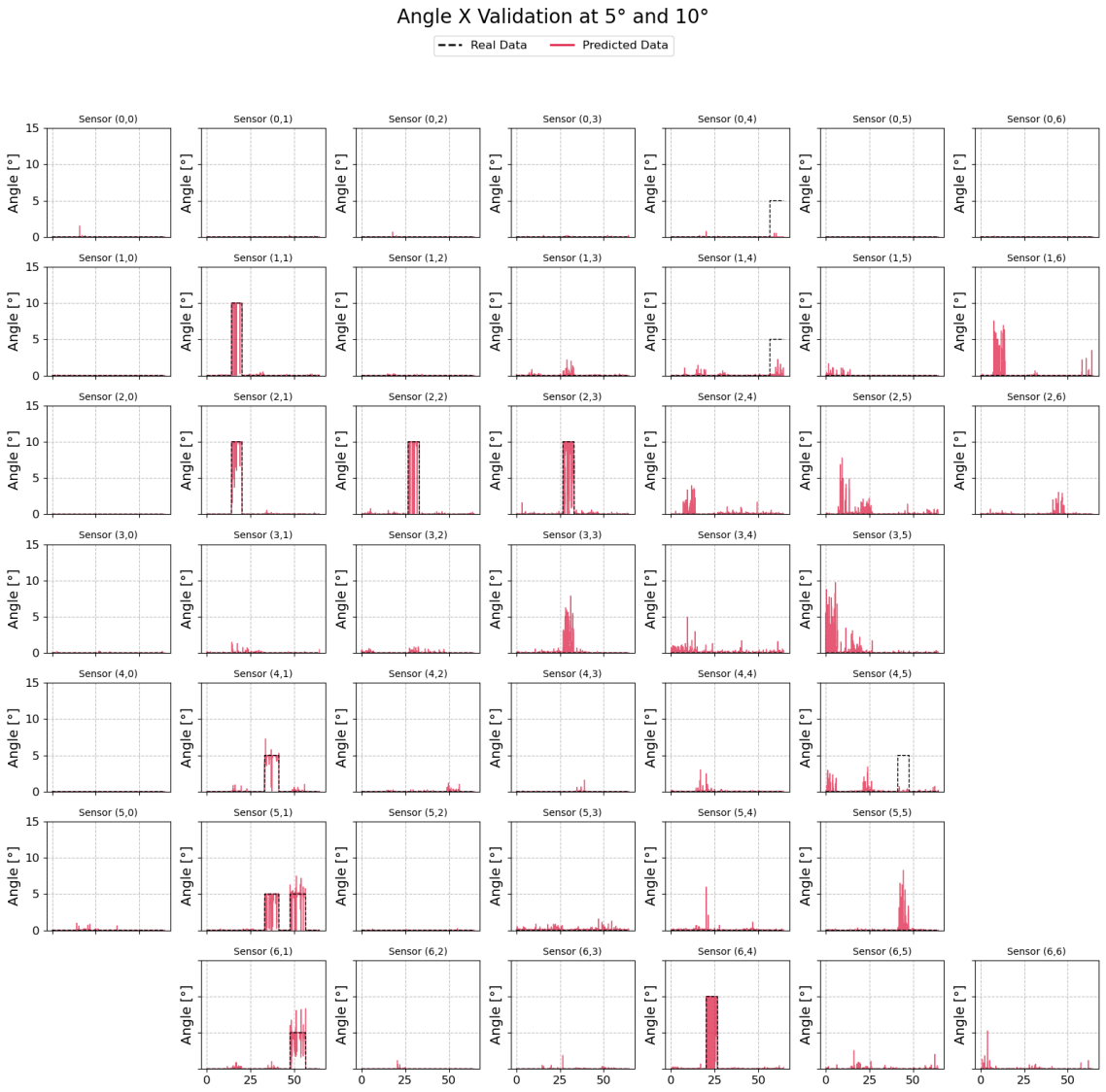


Figure A.9: Estimated and True Force Angles in X across the entire sensor plate, based on readings from each sensor. Force applied at 0° in both angles.

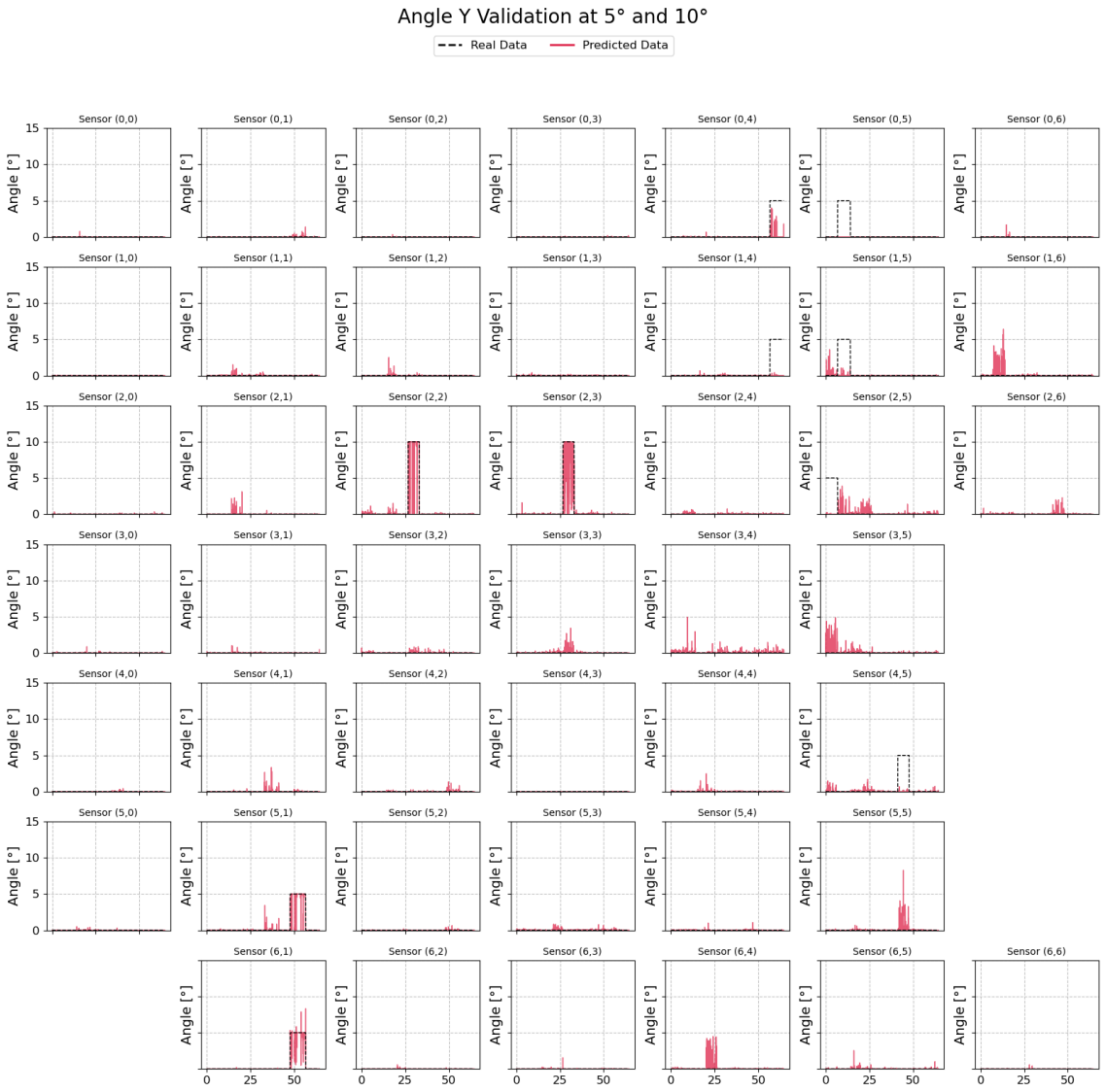
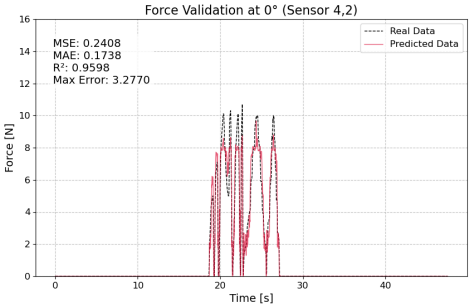
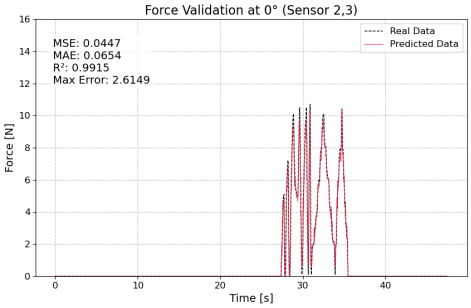


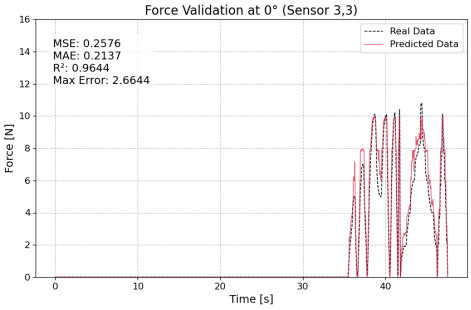
Figure A.10: Estimated and True Force Angles in Y across the entire sensor plate, based on readings from each sensor. Force applied at 0° in both angles.



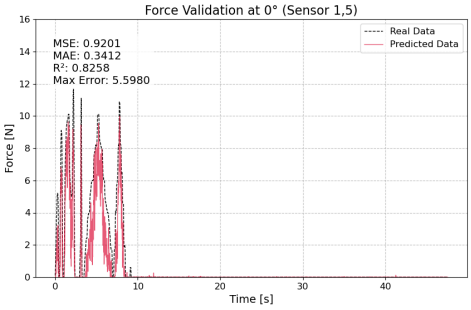
(a) Sensor (4,2).



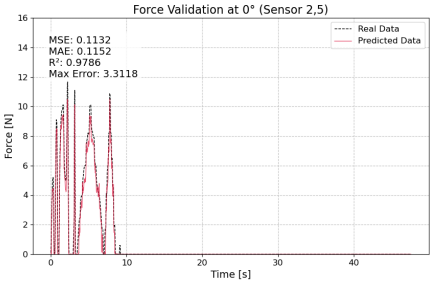
(b) Sensor (2,3).



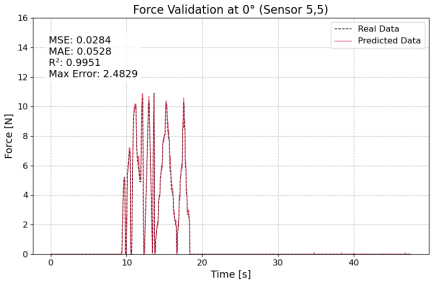
(c) Sensor (3,3).



(d) Sensor (1,5).

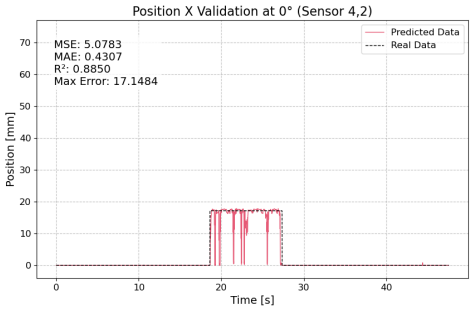


(e) Sensor (2,5)

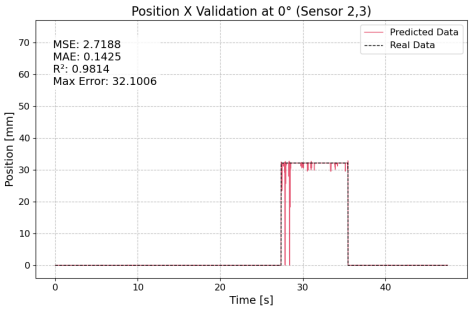


(f) Sensor (5,5).

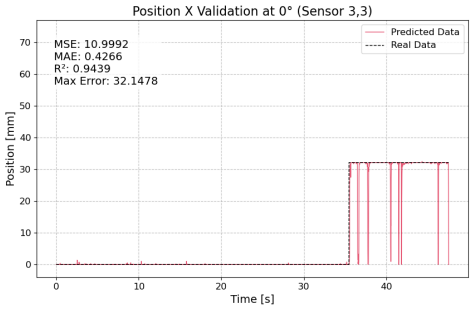
Figure A.11: Comparison of estimated and true forces for specific sensors near the applied forces. Force applied at 0° in both angles.



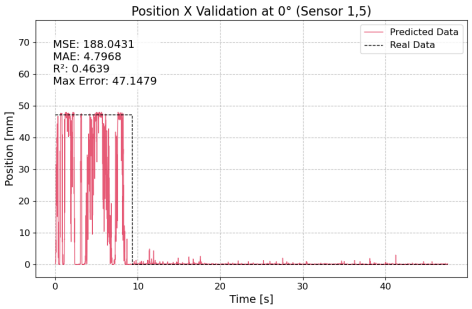
(a) Sensor (4,2).



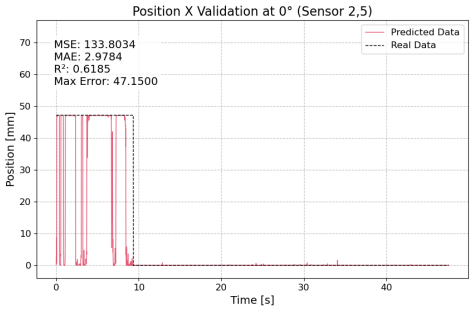
(b) Sensor (2,3).



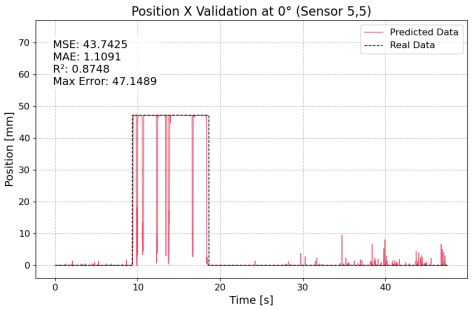
(c) Sensor (3,3).



(d) Sensor (1,5).

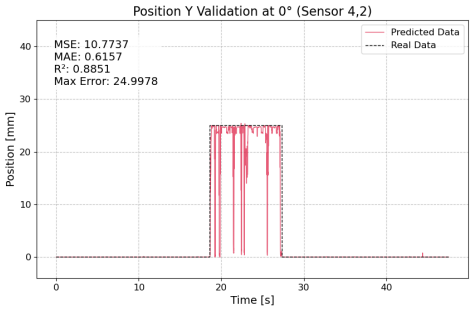


(e) Sensor (2,5)

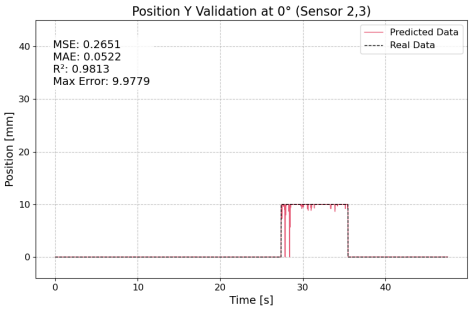


(f) Sensor (5,5).

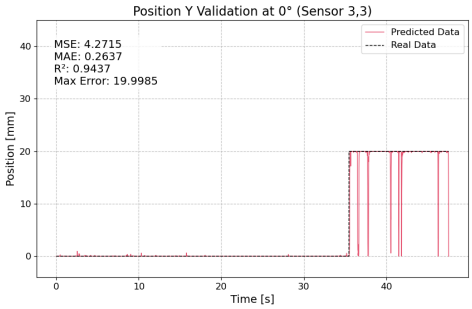
Figure A.12: Comparison of estimated and true Positions X for specific sensors positioned near the applied forces. Force applied at 0° in both angles.



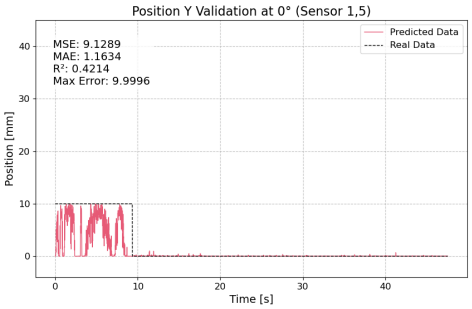
(a) Sensor (4,2).



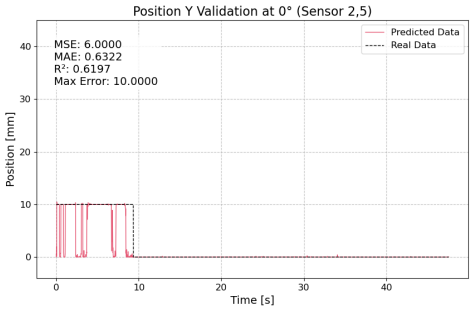
(b) Sensor (2,3).



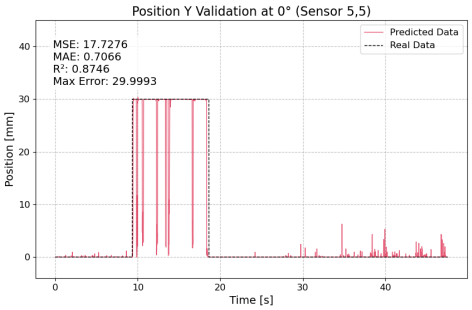
(c) Sensor (3,3).



(d) Sensor (1,5).

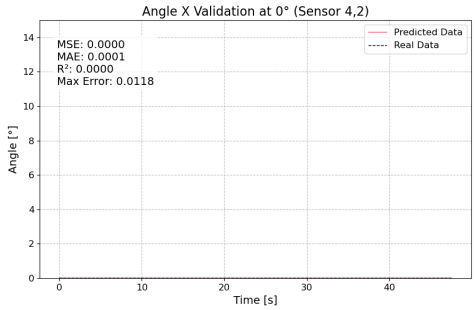


(e) Sensor (2,5)

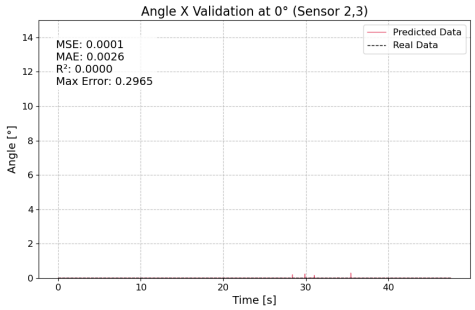


(f) Sensor (5,5).

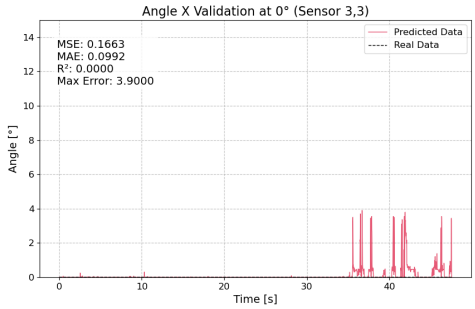
Figure A.13: Comparison of estimated and true Positions Y for specific sensors near the applied forces. Force applied at 0° in both angles.



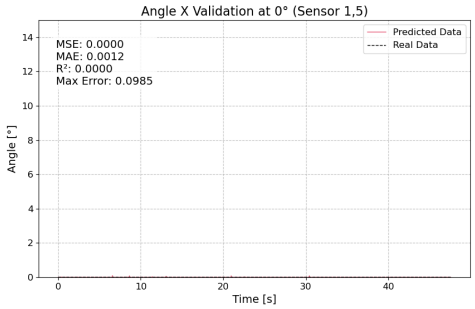
(a) Sensor (4,2).



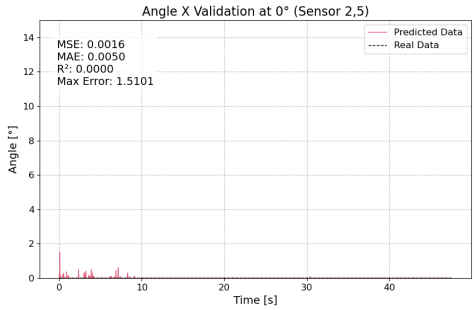
(b) Sensor (2,3).



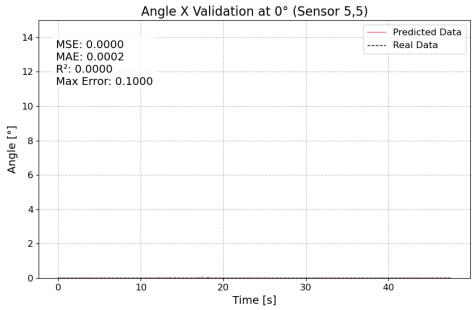
(c) Sensor (3,3).



(d) Sensor (1,5).

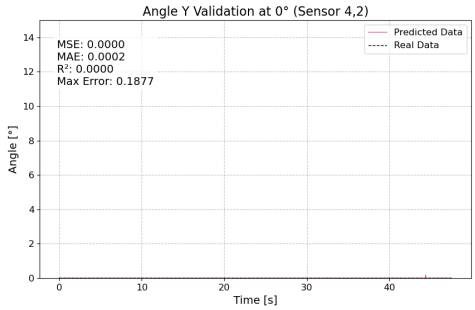


(e) Sensor (2,5)

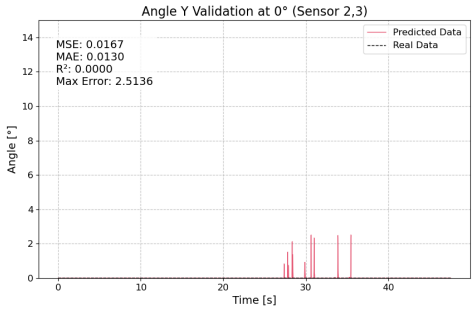


(f) Sensor (5,5).

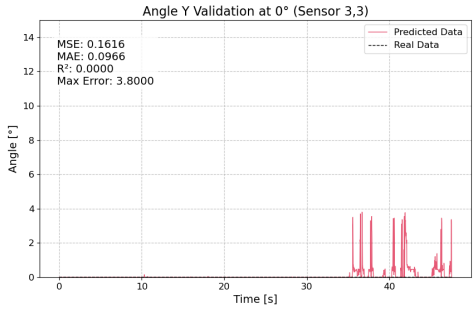
Figure A.14: Comparison of Estimated and True Angles X for specific sensors near the applied forces. Force applied at 0° in both angles.



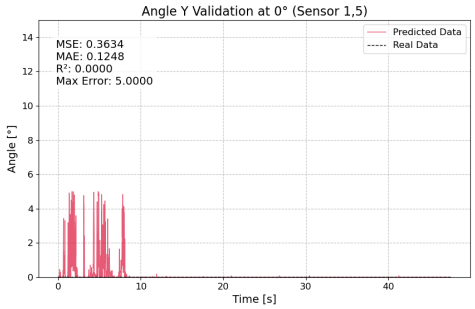
(a) Sensor (4,2).



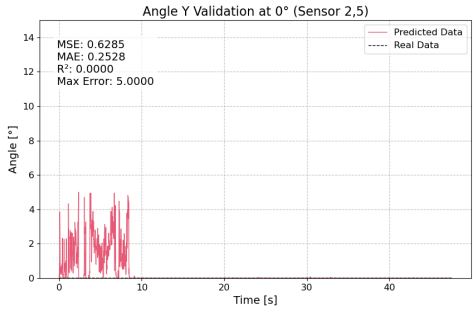
(b) Sensor (2,3).



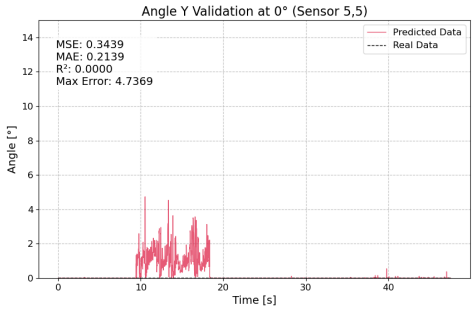
(c) Sensor (3,3).



(d) Sensor (1,5).



(e) Sensor (2,5)



(f) Sensor (5,5).

Figure A.15: Comparison of estimated and true Angles Y for specific sensors near the applied forces. Force applied at 0° in both angles.

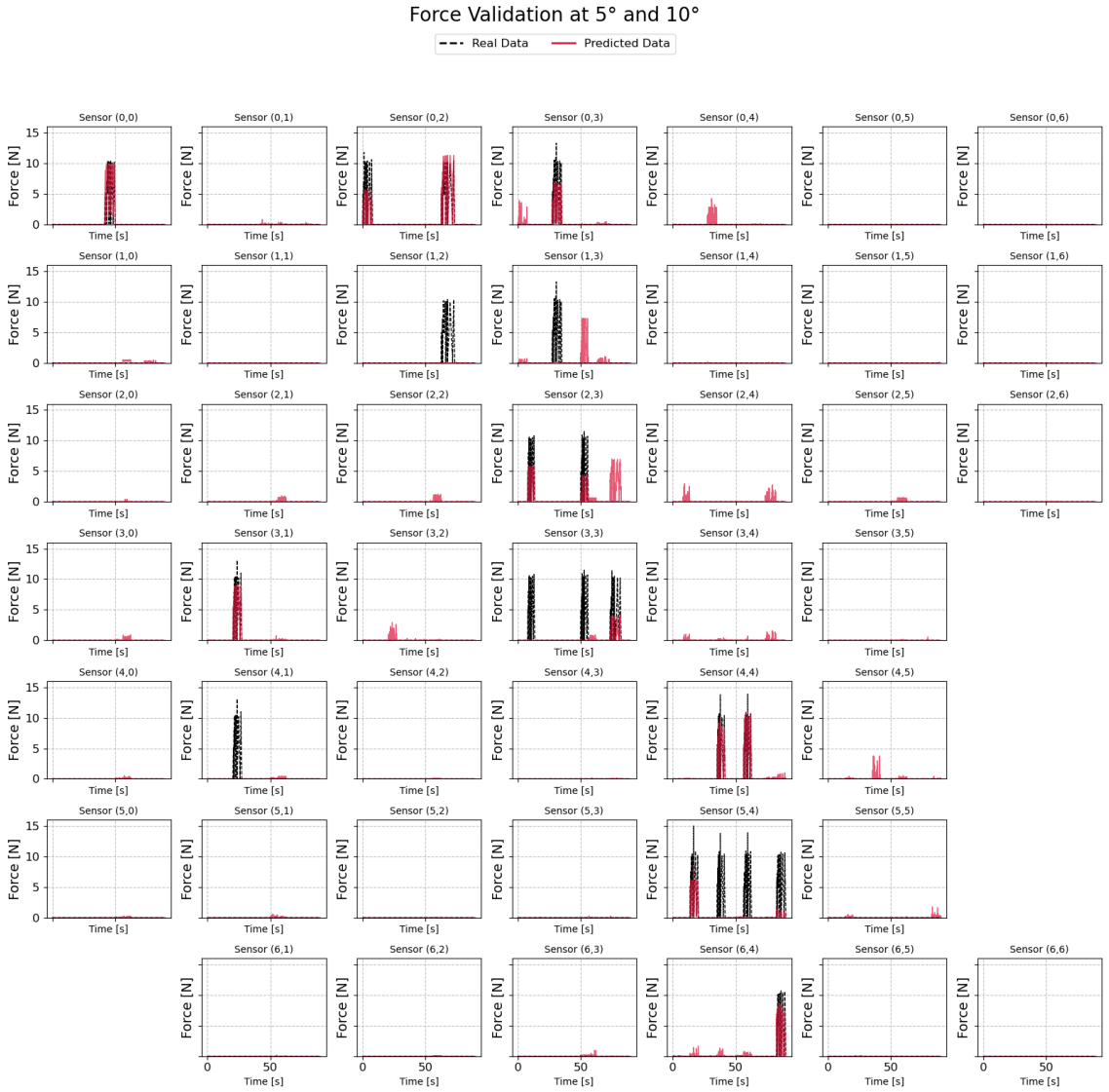


Figure A.16: Comparison of predicted F from each sensor's respective model with actual force readings captured within a 6 mm radius. The figure illustrates how each sensor estimates the applied force, showing both direct readings at the force application point and the distributed predicted force based on readings from surrounding sensors. Force applied at 0° in both angles.

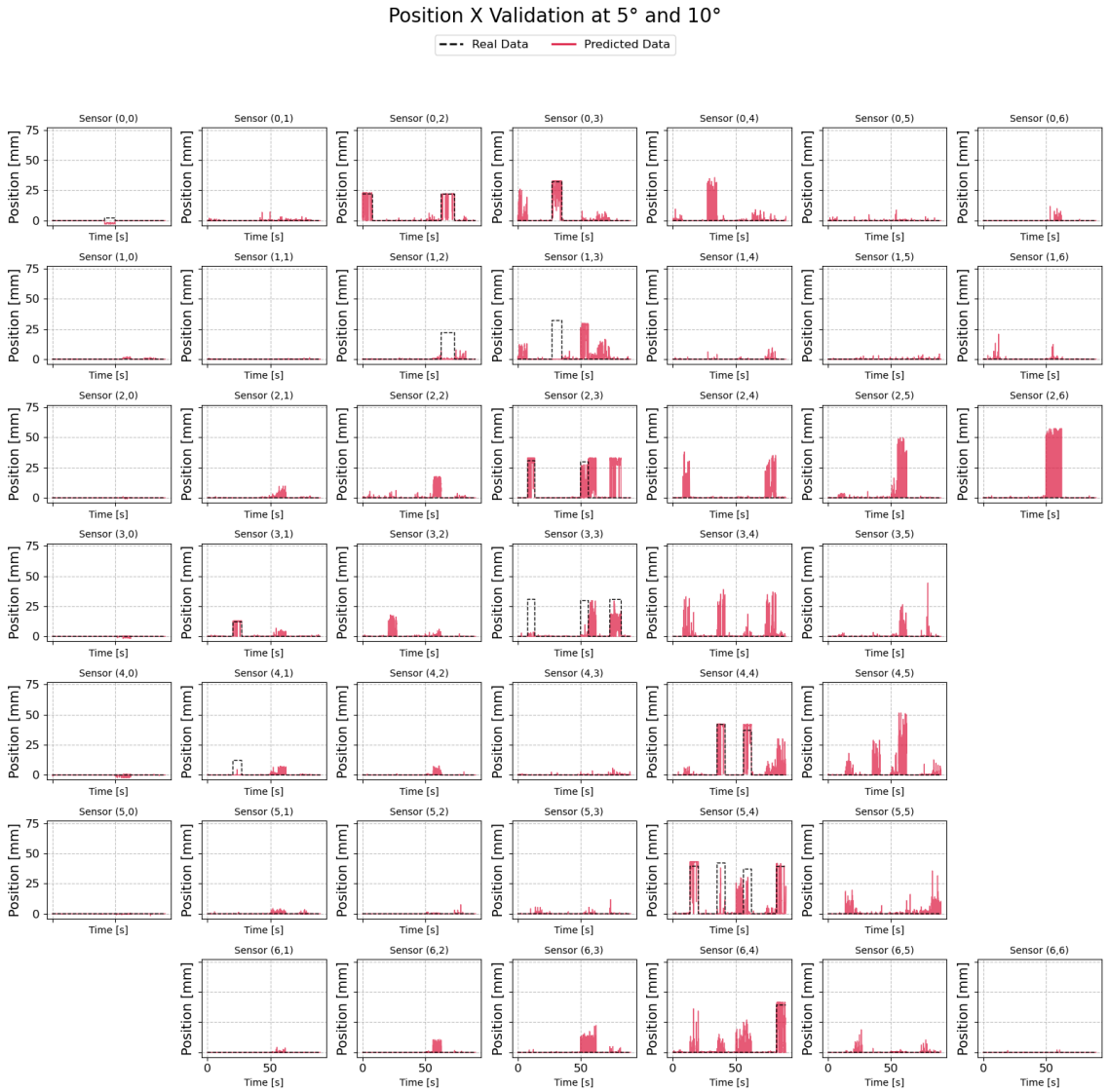


Figure A.17: Estimated and True Force Positions in X across the entire sensor plate, based on readings from each sensor. Force applied at 5° or 10° in at least one of the angles.

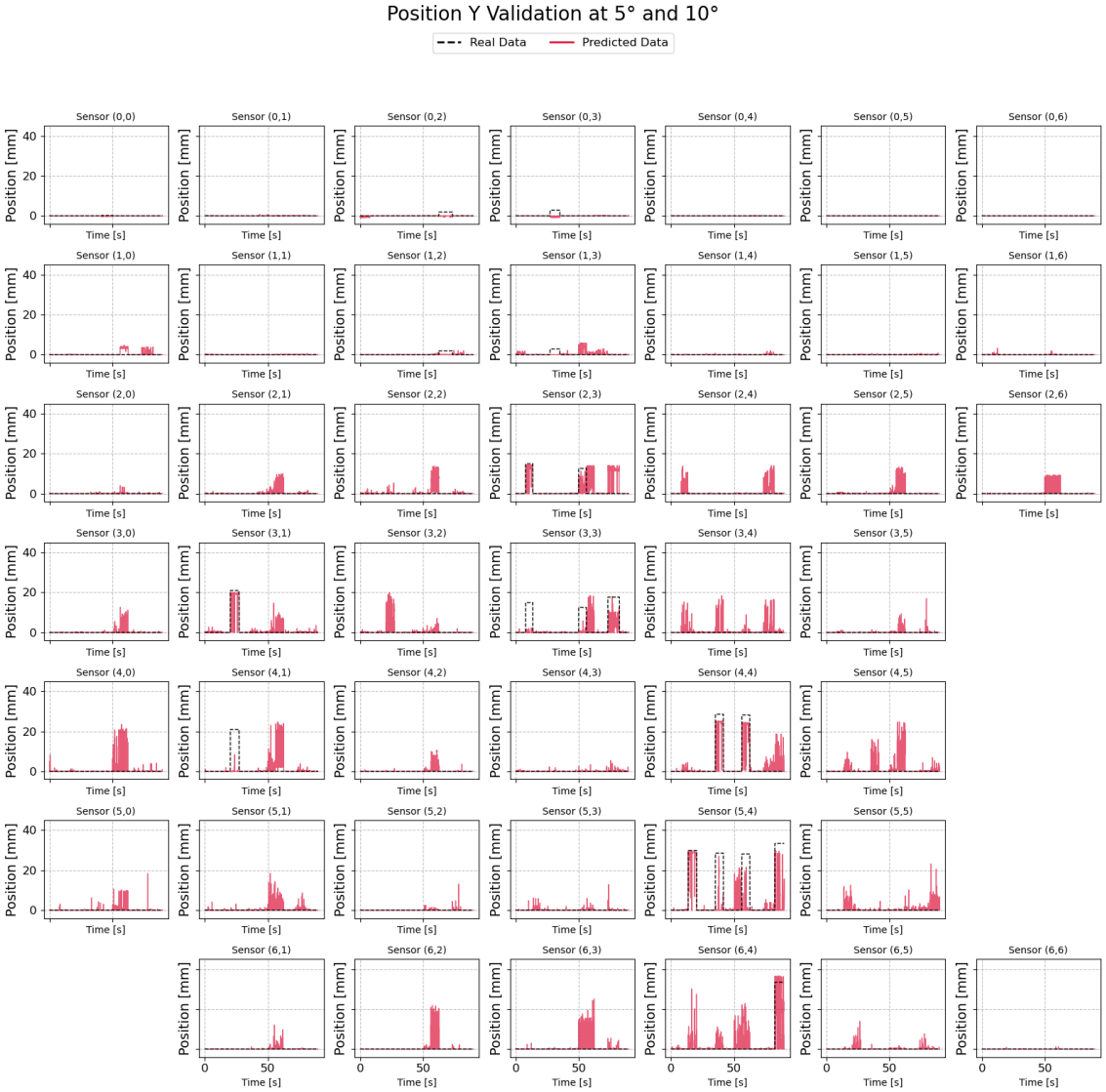


Figure A.18: Estimated and True Force Positions in Y across the entire sensor plate, based on readings from each sensor. Force applied at 5° or 10° in at least one of the angles.

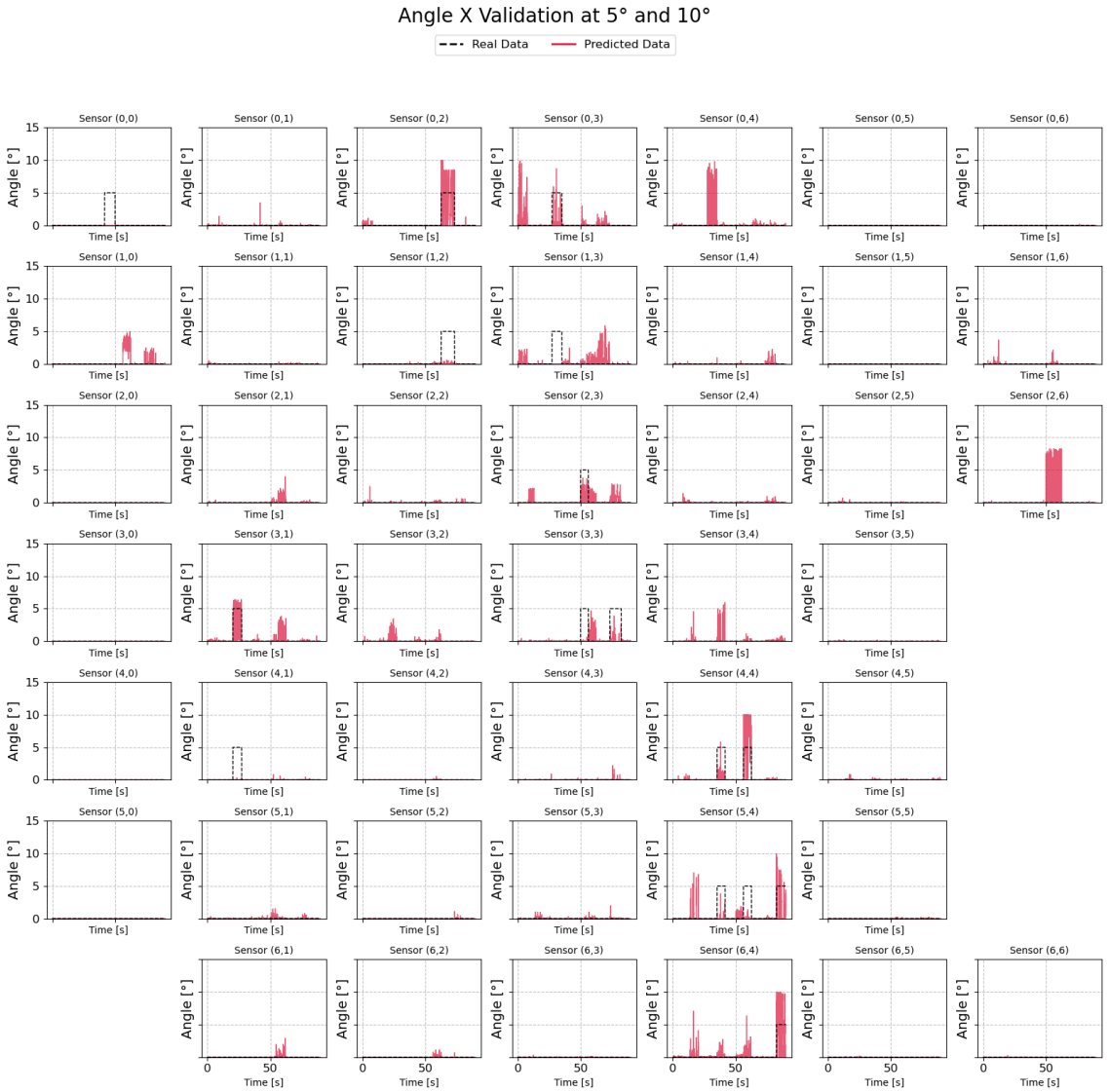


Figure A.19: Estimated and True Force Angles in X across the entire sensor plate, based on readings from each sensor. Force applied at 5° or 10° in at least one of the angles.

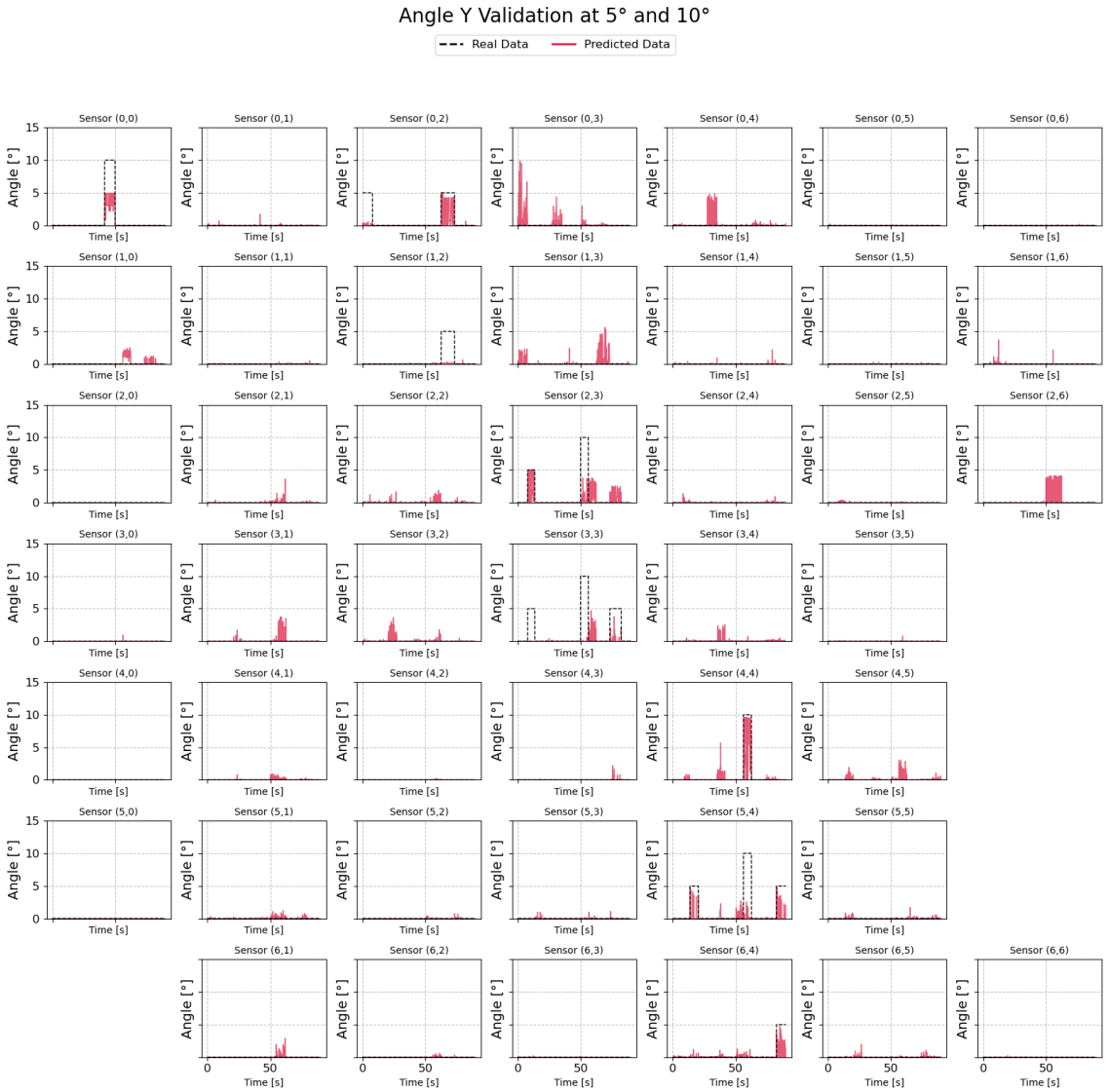
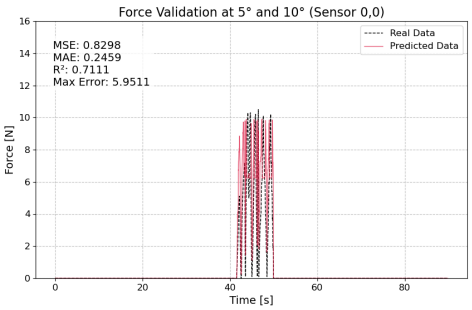
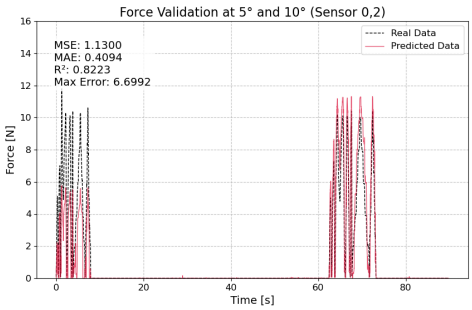


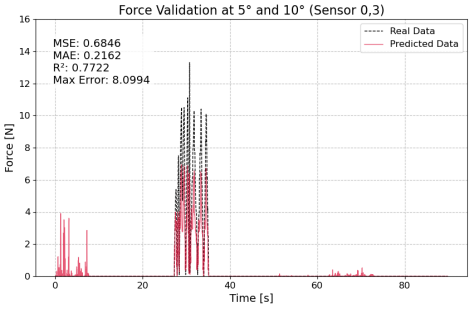
Figure A.20: Estimated and True Force Angles in Y across the entire sensor plate, based on readings from each sensor. Force applied at 5° or 10° in at least one of the angles.



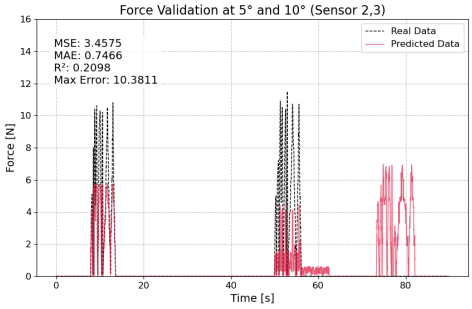
(a) Sensor (0,0).



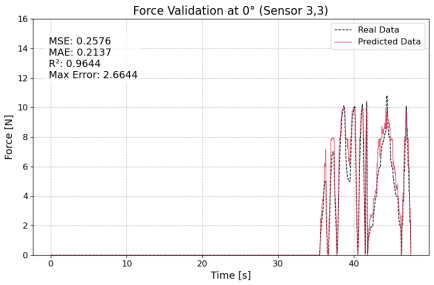
(b) Sensor (0,2).



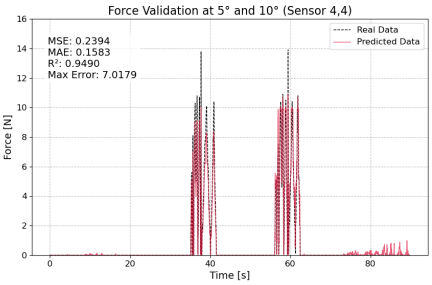
(c) Sensor (0,3).



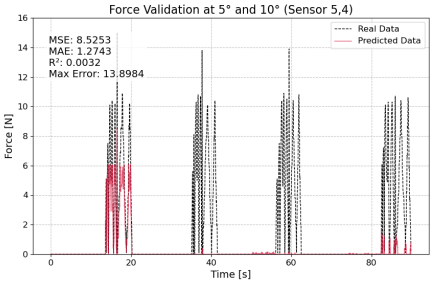
(d) Sensor (2,3).



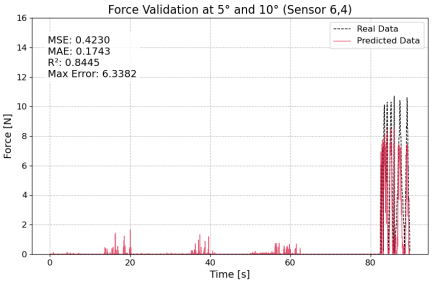
(e) Sensor (3,3)



(f) Sensor (4,4).

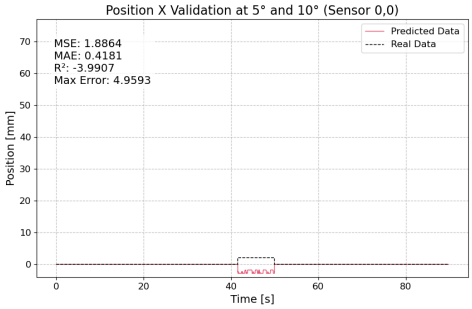


(g) Sensor (5,4)

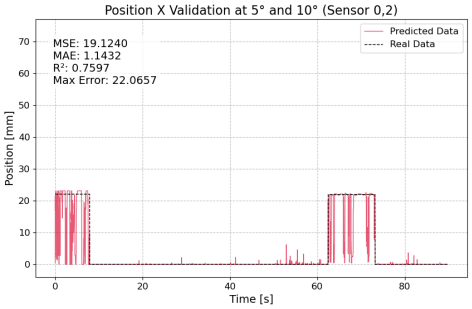


(h) Sensor (6,4).

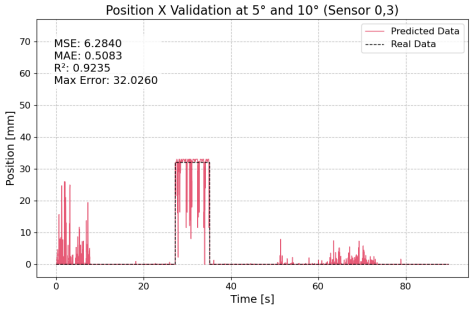
Figure A.21: Comparison of estimated and true forces for specific sensors near the applied forces. Force applied at 5° or 10° in at least one of the angles.



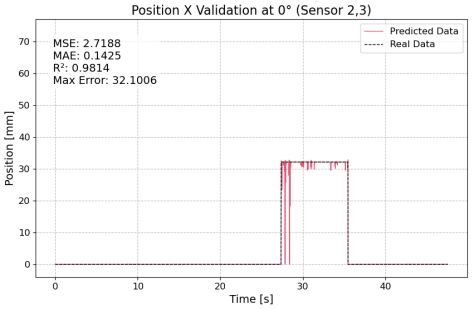
(a) Sensor (0,0).



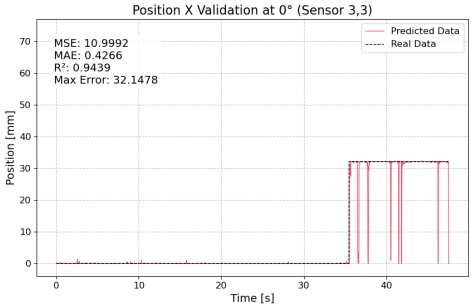
(b) Sensor (0,2).



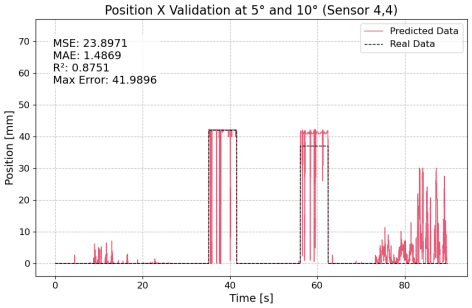
(c) Sensor (0,3).



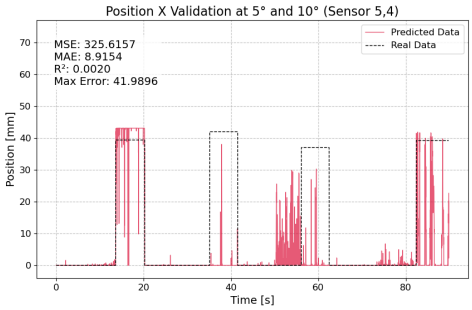
(d) Sensor (2,3).



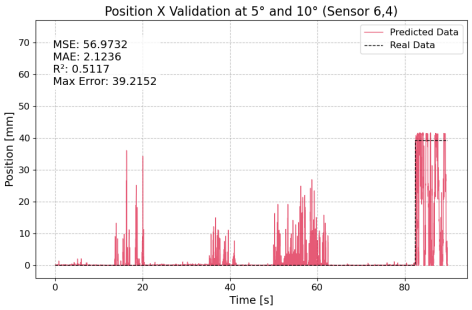
(e) Sensor (3,3).



(f) Sensor (4,4)

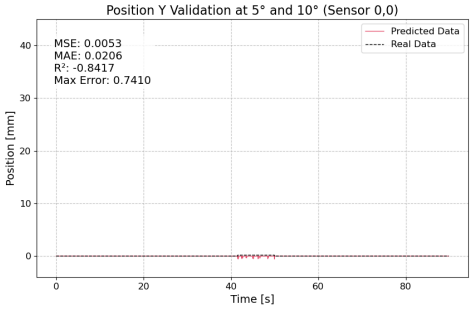


(g) Sensor (5,4).

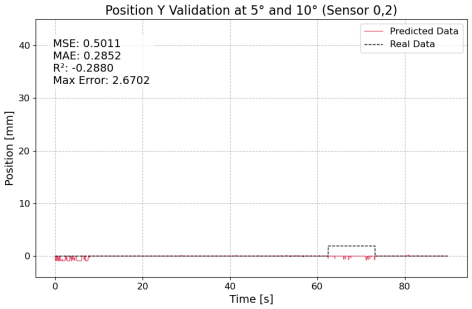


(h) Sensor (6,4).

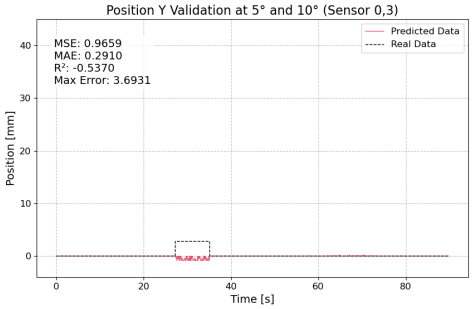
Figure A.22: Comparison of estimated and true Positions X for specific sensors positioned near the applied forces. Force applied at 5° or 10° in at least one of the angles.



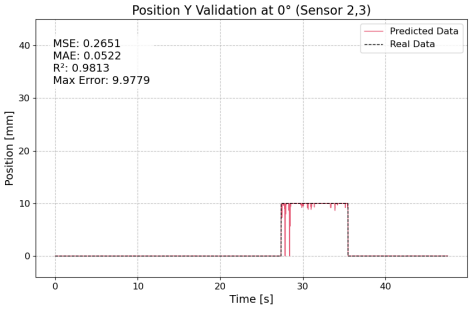
(a) Sensor (0,0).



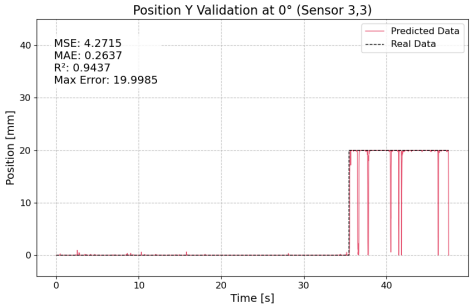
(b) Sensor (0,2).



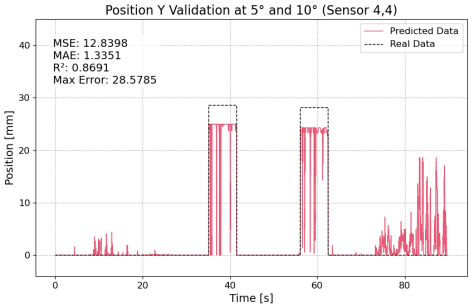
(c) Sensor (0,3).



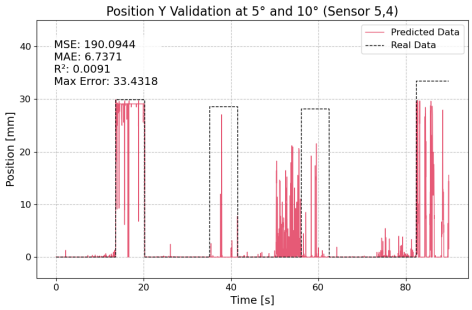
(d) Sensor (2,3).



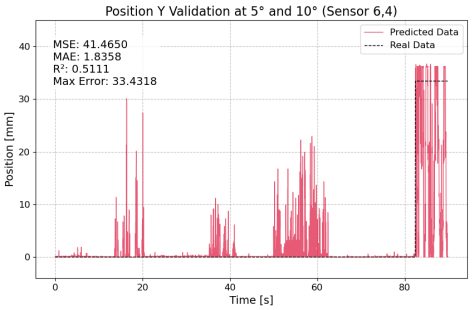
(e) Sensor (3,3).



(f) Sensor (4,4)

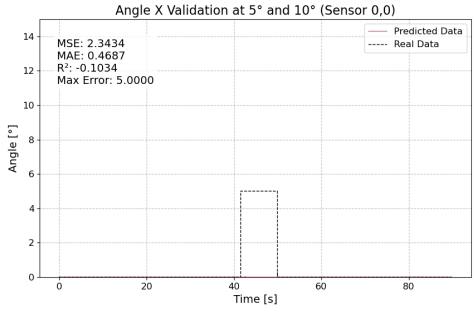


(g) Sensor (5,4).

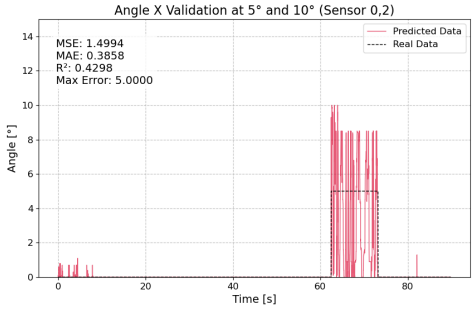


(h) Sensor (6,4).

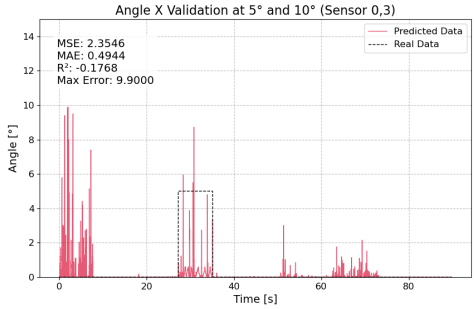
Figure A.23: Comparison of estimated and true Positions Y for specific sensors positioned near the applied forces. Force applied at 5° or 10° in at least one of the angles.



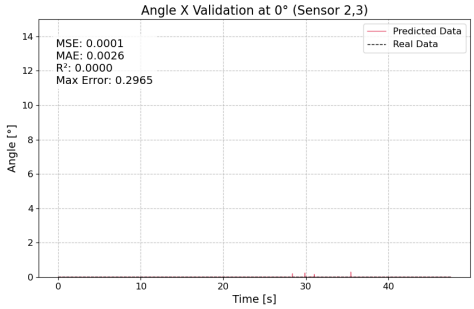
(a) Sensor (0,0).



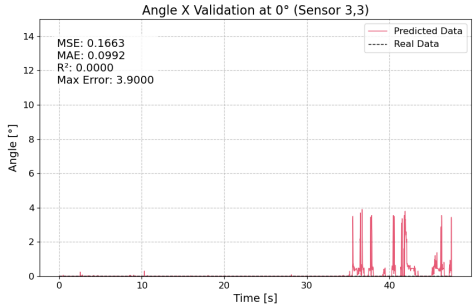
(b) Sensor (0,2).



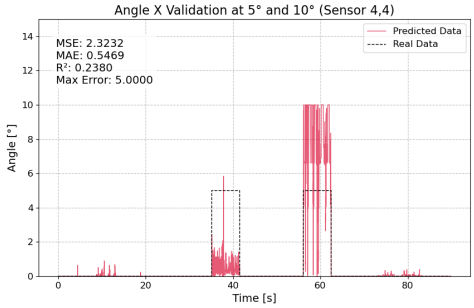
(c) Sensor (0,3).



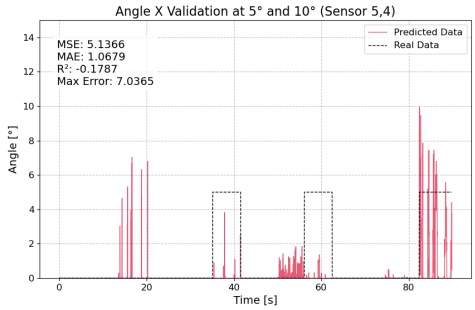
(d) Sensor (2,3).



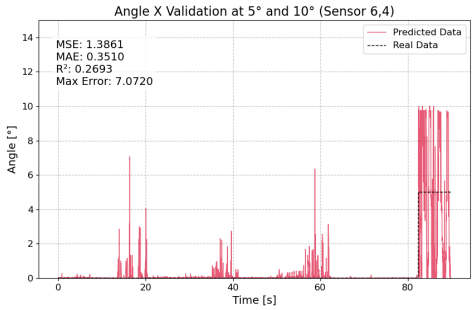
(e) Sensor (3,3).



(f) Sensor (4,4)

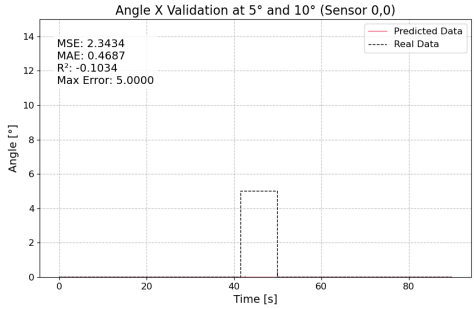


(g) Sensor (5,4).

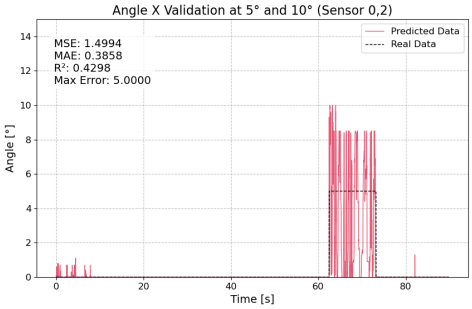


(h) Sensor (6,4).

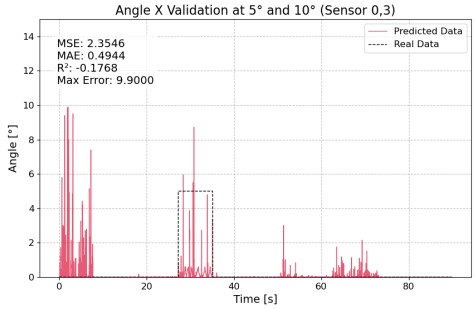
Figure A.24: Comparison of estimated and true Angles X for specific sensors positioned near the applied forces. Force applied at 5° or 10° in at least one of the angles.



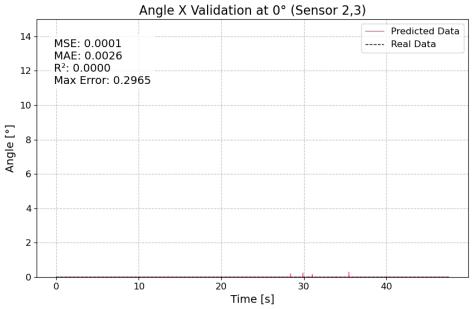
(a) Sensor (0,0).



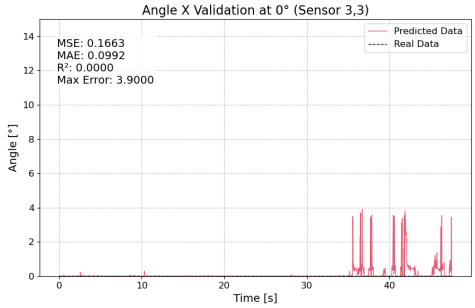
(b) Sensor (0,2).



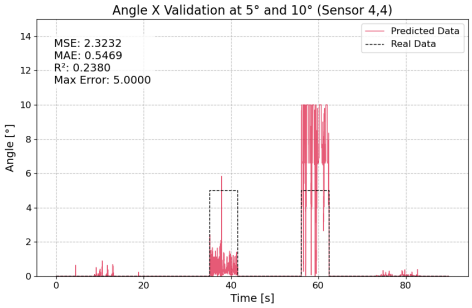
(c) Sensor (0,3).



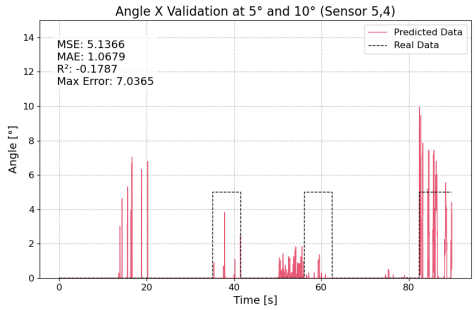
(d) Sensor (2,3).



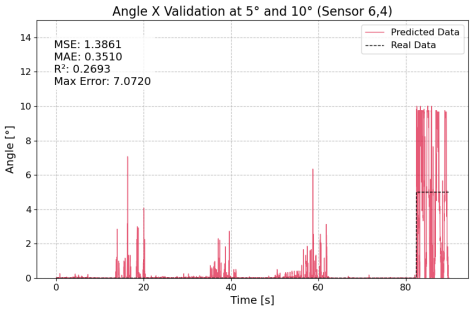
(e) Sensor (3,3).



(f) Sensor (4,4)



(g) Sensor (5,4).



(h) Sensor (6,4).

Figure A.25: Comparison of estimated and true Angles X for specific sensors positioned near the applied forces. Force applied at 5° or 10° in at least one of the angles.

DEPARTMENT OF SYSTEMS AND CONTROL
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY