Low-cost Three Axis Force Plate using Machine Learning

Presented by:
James Teversham

Prepared for:
Dr Amir Patel and Mr Callen Fisher
Dept. of Electrical and Electronics Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Bachelor of Science degree in Mechatronics Engineering

October 22, 2018

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Signature:..
J. Teversham

Date: 22 October 2018

Foreword and Acknowledgments

This project has taken me on an unbelievable learning journey that I feel constitutes the true culmination of what this degree is about. I was initially skeptical of choosing this project due to its novelty and the perceived risk posed by the fact that no student had successfully tried it before me. In the end, this is the very reason that motivated me to spend what now seems like an eternity making sure that everything worked as planned. There was no assurance - even from the project supervisors - that this concept would or even could work and yet, it worked better than my greatest expectations. This would never have been possible if not for the immense support and help that I received along the way and for this, I would like to express my sincere gratitude.

Firstly, to my parents, Kim and Sharon. You've given me every opportunity that I could've asked for in life. I can't express how grateful I am for what you've done for me and for allowing and encouraging me to follow my ambitions. Your support throughout school and my four years at UCT has been nothing short of phenomenal. I dedicate this paper and the inordinate number of hours spent on it to you two.

I would like to thank Mr Callen Fisher, my primary project supervisor and mentor. We both knew mediocrity wasn't an option for this project and your enthusiasm and energy allowed me to push through some seriously challenging times. I'll never forget the coffee you made for me at 7am in the Mechatronics Lab when things were at an all time low. Looking back, there's no one at UCT that I would have rather had watching over my shoulder and I'm so grateful for all the time you spent away from your schedule helping me (and entertaining my jokes which were occasionally[1] even worse than yours). From my experiences in this project, I have every faith that you're going to make a great lecturer!

I would further like to thank Mr Justin Pead who provided me with invaluable practical advice throughout the project. You guided my decision to take a plunge into the unknown by using the AD7730 chips and this project simply wouldn't have worked without them. Your knowledge and time spent helping me with the PCB design is also very much appreciated.

To A/prof Fred Nicolls, I would like to thank you too for our lengthy discussions on how to approach the machine learning aspects of this project. Your guidance and wisdom was indispensable and I've always been astounded by the extent of your knowledge.

Thanks to Becs and Yeshara for your help in testing the plate. Hopefully I'll have lost some weight and won't max out the sensors by the time of my next thesis.

Finally, although there were so many other great people along the way, I'd like to especially thank Chad Reich and Nathan Herr. It's unbelievable to think what we've been through over these four years and I can't imagine what it would've been like without you guys. Whether I was crying from laughter or line-follower-robot-induced depression, you two were always there.

[1] almost never

Abstract

Force plates are widely considered the gold standard in measurement of ground reaction forces (GRFs) in the study of both humans and legged animals. Intricate GRF profiles of human gait measured by force plates are used extensively in the study of human biomechanics. Such information has shed light on some of the most elusive yet prevalent sports injuries such as those of the *anterior cruciate ligament* (ACL). In addition, it has led to extraordinary results in the clinical diagnosis of diseases of the human central nervous system such as Parkinsons Disease and Multiple Sclerosis. Furthermore, force plates have enabled the non-invasive study of animal GRF patterns during rapid and complex manoeuvres which has provided indispensable knowledge towards the design of bio-inspired robots.

Motivated by the extensive utility of force plate technology, the central research challenge addressed by this project is to determine whether a force plate system capable of reliably estimating three axis GRFs can be developed for a few thousand rands. If successful, this system would be the first of its kind to overcome the most significant limitation of existing force plate technology; cost. Existing force plates used in commercial and academic settings are typically priced in excess of $30 000 (USD).

The approach adopted in this project exploits a novel mechanical design that allows three axis GRF data to be captured by only four inexpensive, single-axis load cells arranged in a unique configuration. However, this particular design yields a highly non-linear relationship between sensing elements. Machine learning techniques were employed to circumvent the significant challenge posed by attempting to solve the system analytically.

This project entailed the design and implementation of a complete mechatronic system which comprised of mechanical, electronic hardware and software subsystems (both at an embedded and a high level for data processing and the implementation of machine learning algorithms).

A comprehensive and systematic set of experiments was devised to gather training data for the machine learning models, which were consequently tested and analysed in MATLAB. Even when subjected to highly impulsive three axis forces induced by natural human gait, the system was able to estimate exceptionally well with RMS errors (as % of the full scale force) of 1.71%, 1.54% and 0.57% for the x-, y- and z-axis GRF components respectively. Remarkably, three axis GRF profiles recorded by the system correlated extremely closely to those recorded in published studies of human gait presented in [9] and [11].

A central conclusion of this study was that, with improved training techniques and refinement to its design, the force plate system developed could forseeably pose a viable alternative to existing force plate technologies in applications previously mentioned. Excitingly, this entire system prototype was produced at a cost of approximately 50 times less than a typical commercial model and suggestions are made to reduce this cost in future work.

Contents

List of Figures

List of Tables

Glossary and Acronyms

- CNS: Central nervous system
- PD: Parkinson's Disease
- MS: Multiple Sclerosis
- RMS: Root-mean-square
- ADC: Analogue-to-digital converter
- DAC: Digital-to-analogue converter
- IC: Integrated circuit
- ODR: Output data rate
- PGA: Programmable gain amplifier
- DMA: Direct memory access
- FIR: Finite impulse response
- MEMS: Micro-electro-mechanical systems
- FEM: Finite element method
- MCU: Micro-controller unit
- FPU: Floating point unit
- APB: Advanced peripheral bus
- PCB: Printed circuit board
- FIFO: First in, first out
- F/T sensor: Force/torque sensor
- GPIO: General purpose input output
- Rx: Receiver (or receive)
- Tx: Transmitter (or transmit)

Chapter 1

Introduction

1.1 Motivation and Background to the Study

The use of force plates in industry and academia is extensive. In studies of human biomechanics, force plates have proved invaluable in understanding the mechanics of sports injuries as well as the kinematics of exercise [1]. Many clinical studies [9, 10, 8] have used force plates and machine learning methods to analyse kinetic features of human gait. This has been instrumental in the diagnosis of motor-related diseases of the human central nervous system (CNS) such as Parkinsons Disease and Multiple Sclerosis with up to 98.2% accuracy [9].

Another exciting field of study is that of Biomimicry and bio-inspired robotics. For some time, human technology was seen to be in stark contrast to nature - almost two mutually exclusive fields; us, the humans, and everything else. However, humans have become increasingly intrigued in how animals exploit some of the most incredible mechanisms to perform complex manoeuvres and achieve superhuman dexterity and agility. The cheetah (*Acinonyx jubatus*) has attracted much interest in academia owing to its phenomenal agility and speed. Testament to this are MIT's Cheetah quadruped robots [13] and Boston Dynamics' WildCat quadruped robot, as well as the extensive studies on the cheetah by UCT's very own Mechatronics Laboratory [17, 18, 19, 20]. However, measuring GRFs produced by a cheetah rapid dynamic manoeuvres has hardly been studied due to the extreme difficulty in doing so. Although several types of three-axis force sensors exist, they are typically extremely expensive and cannot be physically attached to the animal without interfering with its capacity for uninhibited motion.

Force plates offer a non-invasive method for achieving this. However, multiple stationary force plates would be required to gather sufficient data during the cheetah's motion. This leads to the most significant limitation of existing commercial force plates; cost.

Development of a low-cost three axis force plate that can accurately estimate GRFs at high speed will allow several force plates to be deployed in the study of the cheetah - potentially even for the fraction of the price of a single commercially available version. Such a platform could also be of great use to the Sport Science Institute at UCT and beyond in applications mentioned above.

1.2 Aim of the Study

The aim of this study was to test a robust and low-cost[1] force plate capable of estimating three axis GRFs.

1.2.1 Objectives of the Study

The fundamental goal of this project was to create a fully-fledged mechatronic system developed from scratch and in doing so, gain knowledge and exposure to a wide range of skills. The project objectives are outlined below:

1. Review literature pertaining to the applications and technology behind force plates. Perform a brief review of other mechanisms for GRF estimation and comment on their effectiveness compared to force plates.
2. Review the mechanical design of the plate proposed by the project supervisors and perform simulations to verify the mechanical integrity of the design
3. Design and implement the electronic hardware and embedded software required to sample data from all relevant sensors with high fidelity and speed
4. Design and implement a sensing mechanism for determining the position of the applied force on the plate
5. Design a data packetisation and integrity-checking scheme to log sampled data to a permanent storage medium
6. Implement software for extraction of logged data and processing thereof
7. Devise and conduct a series of rigorous loading tests on the plate to gather extensive training data with which to train machine learning models
8. Critically evaluate and analyse the performance of machine learning models in estimating GRFs

1.2.2 Significance of the Study

The prospect of a three axis force plate produced at a *significantly lower cost* than commercially available versions is extremely relevant to both commercial and academic applications.

[1] In the region of a few thousand rands.

According to [1], while highly desirable in sports science applications for providing insight on human gait patterns and force characteristics during various manoeuvres s mentioned above, high-end force plates are rarely available in private practices due to their exorbitant cost. A low-cost platform would be indispensable to private practices, smaller businesses and educational institutions (such as schools and universities) where the cost is a significant constraint.

Studies such as [8, 9, 10] and several others have shown how force plates can be used in clinical diagnosis of motor-related CNS disorders. A low-cost force plate would allow many other similar studies to performed and would potentially advance the understanding of these diseases at a more rapid rate as a result.

A low-cost force plate array would be invaluable in studies of the cheetah and other quadruped animals and would provide insightful information to researchers in the development of bio-inspired robots. In particular, this data would be of great use in the study of the cheetah being conducted by the UCT Mechatronics Laboratory.

1.3 Scope and Limitations

Although the intention of this project was to develop a fully-fledged system with all the hardware and software required for an operational force plate, the project forms more of a proof-of-concept than a fully tested and finished product. It is intended to demonstrate whether or not the novel mechanical design proposed by the project supervisors can indeed be used to create a low-cost force plate, and if so, the extent to which GRFs can be accurately estimated using machine learning methods.

Unfortunately, this platform could not be tested on wild animals as the ethics clearance required to do so was not acquired. Furthermore, practical considerations pertaining to experiments performed on cheetahs or other animals - such as the method used to ensure that the cheetah will actually step on the force plate - are not included in the scope of this project.

A detailed static and dynamic mechanical analysis of this system was deliberately excluded from the scope of this project as a core objective of this project was to investigate leveraging machine learning techniques as an alternative to solving mechanics of the system analytically.

1.4 Research Questions to be Investigated

The following research questions were posed and answered during this study:

1. Can a low noise, high fidelity data acquisition system capable of sampling at 1kHz be developed and used reliably?

2. Is it possible to use four single-axis load cells mounted in the particular configuration outlined in Section 5.1 together with machine learning techniques to accurately estimate three axis force sensors for under R2000?

3. Is it foreseeable that the low-cost force plate system developed in this project could serve as a viable alternative to high end industrial versions used in the study of humans and animals?

1.5 Plan of development

This report begins with a thorough review of literature pertaining to the technology and applications of force plates in Chapter 2. A review on bio-inspired robots and an investigation into other technologies used to estimate GRFs is included in this chapter. As these concepts were not covered in university course work and presented the greatest theoretical challenge, important theory pertaining to the machine learning models utilised in this project is explored in the Theory Development in Chapter 3.

Chapter 4 details the methodology and experimental procedure followed throughout the various stages of this project. This chapter details the procedures used to generate training and testing data for experiments used to test performance of machine learning models.

The design of electronic hardware and embedded software comprising the data acquisition system is expounded in Chapter 6 and Chapter 7 respectively. The practical implementation of machine learning models in MATLAB is documented in Chapter 8. Post-processing software used for extracting and decoding log data is also documented in this chapter.

The performance of the designed system is quantified in Chapter 9 and thereafter, a discussion of the results obtained is presented in Chapter 10. Finally, conclusions are drawn based on the results and subsequent discussion and recommendations for future work are provided in Chapter 11. The layout of this report is summarised in Figure 1.1.

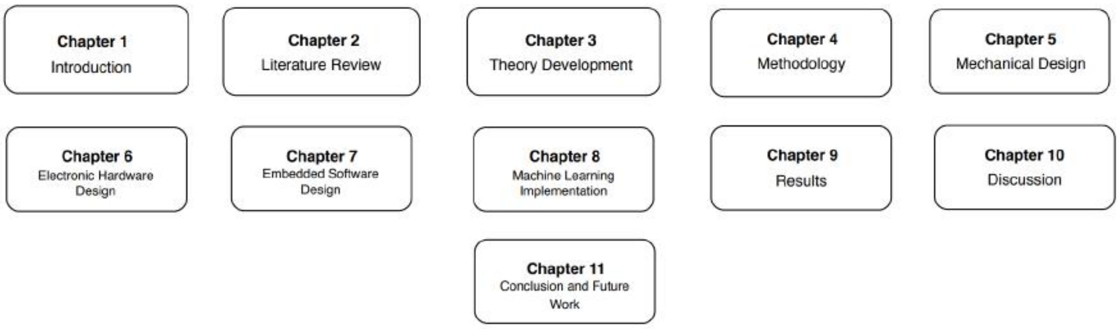


Figure 1.1: Diagram showing a summarised layout of this report

Chapter 2

Literature Review

While force plates have been around for some time and have been studied fairly extensively, their utility in new fields continues to expand. Force plate systems are commonly used to measure multi-axial ground reaction forces (GRFs) used in biomechanical analysis of gait an various other motor activities in humans [1]. Furthermore, in aiding studies of biomimicry and bio-inspired robotics, force plates offer a highly effective means of studying GRF interactions experienced by animals during high speed transient manoeuvres as they provide a non-invasive means of capturing data.

Following a brief explanation of GRFs, a review of the academic and commercial applications of force plate technology is presented, including literature pertaining to the study of humans and animals alike. Thereafter, insight is provided into some of the other mechanisms of estimating GRFs. A refined investigation is then conducted on the specific sensing technologies used in force plates and a review of the current state-of-the-art in this technology. Finally, a summary of the findings of this chapter is presented in Section 2.6.

2.1 Ground Reaction Forces

A GRF can be considered a three dimensional vector that characterizes the force exerted by the ground on a body in contact with it. More specifically, it represents the resultant reaction force of the distributed load applied the body interacting with the surface (as depicted in Figure 2.1). The three orthogonal vectors that form the resultant GRF are measured in the vertical (z), longitudinal or anterior-posterior (y) and lateral or mediolateral (x) axes. The centre of pressure (COP) is defined as the point of application of the GRF vector (in the xy plane).

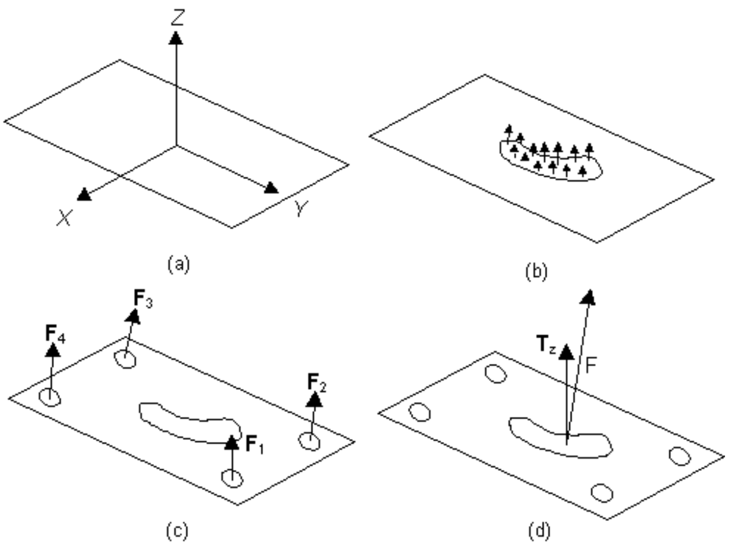


Figure 2.1: Diagram illustrating the resultant GRF (F) and CoP of a distributed load (frame (b)) measured by a force plate [60]. The free torque induced (T_z) is also indicated.

2.2 Biomechanics of Physical Rehabilitation in Humans

2.2.1 Physical Injury

GRFs have been used extensively to quantify anomalous limb loading in humans in order to gain greater insight into internal joint stress and potential causes and indicators of injury [2]. The combination of large, impulsive forces and the high frequency of absorbing/exerting force during landing in sport produces significant stress in joints in the lower limbs [3].

The vertical GRF (VGRF) experienced by a subject during landing is frequently used to determine parameters such as the maximum force and time to maximum force exerted. These parameters provide an indication of the intensity and duration of stress experienced on a subject's limbs and internal joints [2]. Importantly, large and impulsive VGRFs have been correlated to an appreciable increase in the risk of sustaining an *anterior acruciate ligament* (ACL) injury - a particularly serious injury prevalent in many sports. More specifically, a study by Hewett *et. al.* [5] showed that peak VGRFs were, on average, 20% larger in females who had sustained ACL injuries compared to uninjured individuals [2]. Furthermore, Impellizzeri *et. al.* showed that testing GRFs (and most importantly, the VGRF) in jumping and landing techniques has proved a reliable method for assessing bilateral strength asymmetry [6]. It was shown that observing asymmetries and other abnormalities in the force profile of the left and right legs during landing could be used to identify potential injury risks, as well as the ability to predict recurring injury (such as tearing in ACL reconstructions) [7].

2.2.2 Force Plates and Athletic Performance

Force plates are particularly useful for evaluating kinetic behaviour of an athlete's movement by providing information about external forces involved in their movement. GRFs are often used to provide a quantitative evaluation and means of tracking an athlete's development or ability to execute a particular skill [1]. Furthermore, studies show that GRFs and metrics derived from them provide significant correlation to agility and sprint performance [1] and provide insight that mere visual cues may not.

2.2.3 Clinical Diagnosis of Human CNS Disorders

Posturography

Posturography incorporates the techniques used to quantify control of posture in an upright stance in either static or dynamic conditions with the aid of a force plate [8]. This technique is used as a specialized method of clinical assessment of feedback mechanisms of the human CNS in the control of posture and balance. This technique is both applicable to healthy individuals, where it is used in physical education and sporting analysis, as well as those who suffer from medical disorders with which impairments in gait and balance are associated. Multiple sclerosis (MS) is one such disorder that causes damage to the CNS of the patient, leading to impairments in postural response and capacity to maintain balance [8]. Increased fall tendency and balance impairment are frequently observed in patients with MS. It is often desirable to quantify these characteristics in trying to reach a diagnosis in patients where there is a suspicion of motor disorder following an injury or disease that has lead to a potentially compromised CNS.

Static posturography is commonly used to assess the postural sway of the centre of pressure (COP) of the patient - the point of application of the resultant GRF exerted/experienced by the patient). Metrics such as the velocity of the COP in the lateral (x and y) axes, the sum of displacements of the COP and the 95% confidence ellipse area of the COP are commonly used in such assessments. Static posturography provides a linear, objective and reliable quantitative assessment of static balance [8] and from a more qualitative perspective, the displacement of the COP represents an indicator of the energy expenditure required to maintain balance [8].

Human Gait Analysis

Another extensively used technique in diagnosing motor-related disorders in humans is gait analysis. Tahir et al [9] explored the use of machine learning techniques to classify patients with Parkinson's Disease (PD) by observing gait patterns obtained with force plate data. In particular, the use of artificial neural network (ANN) and suport vector machine (SVM) classifiers (as explored in Chapter 3) were used. Motor-related PD is a degenerative illness of the brain that hampers the patient's ability to execute simple motor tasks such as walking and balancing [10]. It has been shown that motor PD significantly affects a patient's gait by reducing speed, stride length and range of movement during walking and power generation during push off in the ankle power profile [9]. The aforementioned study used the following kinetic features in attempting to classify (diagnose) patients with PD by distinguishing them from healthy patients:

- Maximum vertical heel contact force (F_{z1})
- Minimum vertical mid-stance force (F_{z2})
- Minimum vertical push-off force (F_{z3})
- Maximum horizontal heel strike force during braking (F_{y1})
- Maximum horizontal push-off force (F_{y2})

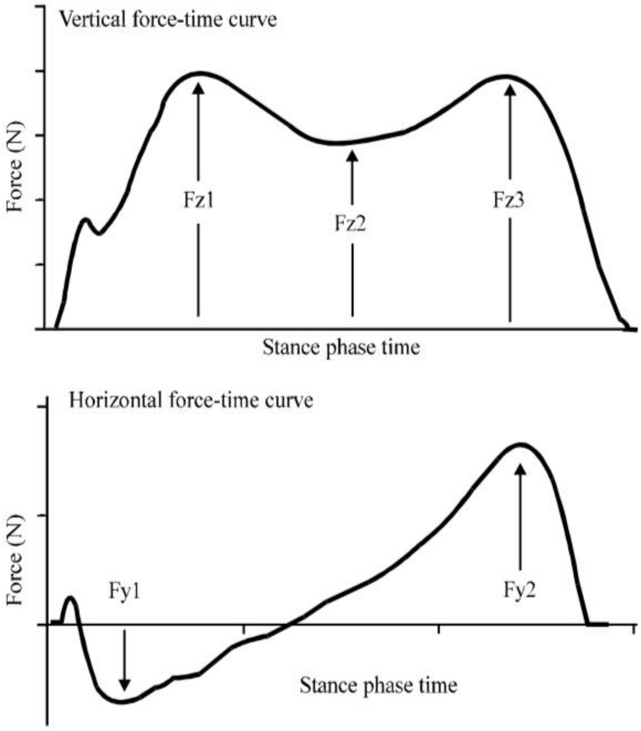


Figure 2.2: Graphs to show force-time curves for vertical (F_z) and anterior-posterior (F_y) GRF components for experiments conducted in [9]. The kinetic features listed above are indicated on the curves.

Results showed that a SVM regressor with a polynomial kernel and a feed-forward ANN achieved 95.8% and 90.6% classification accuracy respectively based on the above kinetic features alone. Fusion of the kinetic features with basic spatiotemporal features such as stride time, cadence (steps per minute), step length and walking speed allowed for the highest accuracy of 98.2% using an SVM with a Gaussian radial basis function (RBF) kernel [9]. It is interesting to note, however, that using force information from only *two axes* (z and y) in the kinetic feature tests allowed for a classification accuracy of very close to the maximum.

A second study by LeMoyne *et. al.* [12] showed that in attempting to classify muscle history dependence in patients with a particular prosthesis, the same kinetic features as with [9] were studied - indicative of the usefulness of these features. Results in this study showed that when incorporating both kinetic and temporal features, classification accuracy increased from 85% when using purely kinetic features to 91% with the fused data set. Again, it is worth noting that only two components (vertical and anterior-posterior) of the GRF vector measured by the force plate were used in this classification study.

Following the discussions above, it is evident that the use of force plates in the study of human biomechanics is extensive. However, the utility of force plates is not limited to the study of humans alone; their ability to measure GRFs in an extremely accurate and non-invasive manner has contributed to their popularity in the study of legged animals. As in the study of cheetahs in [22], force plates have facilitated the acquisition of information such as the stride time, force profiles and the distribution of body weight of animals during rapid manoeuvres. Such insight offered by force plates has been indispensable in the design and development of bio-inspired robotics and the field of biomimicry as explored in Section 2.3 below.

2.3 Force Plates and Bio-inspired Robotics

Many of the most complex manoeuvres performed by legged animals are initiated through interactions with the ground. Importantly, detailed knowledge of GRF profiles induced during hopping, jumping, running and landing of animals that can be acquired using force plates presents invaluable information to researchers in the design of bio-inspired robots. In [16], a force plate was used to investigate how domestic cats are able to manipulate their landing from various heights. It was found that cats adopt measures to deliberately extend the time of impact when landing in order to reduce the force experienced. This can be explained by the equations 2.1 and 2.2 relating the force experienced by a body, $\vec{F}$, to its change in momentum, $\Delta\vec{p} = m\Delta\vec{v}$:

$$J = \int_{t_1}^{t_2} \vec{F}\,dt = \int_{t_1}^{t_2} m\vec{a}\,dt \tag{2.1}$$

and if $\vec{F}$ and $\vec{a}$ are constant:

$$J = \vec{F}\Delta t = m\Delta\vec{v} \implies \vec{F} = \frac{m\Delta\vec{v}}{\Delta t} \tag{2.2}$$

where Δt is the duration of time that the body is subjected to force $\vec{F}$ and $\vec{a}$ and $\vec{v}$ represent the body's resulting acceleration and velocity respectively. Note that $J = \Delta\vec{p}$. That is, the impulse J imparted by a force on a body is equal to that body's change in momentum. Furthermore, [16] reported that the force plate data showed *four distinct spikes* in the measured VGRF profile. The author theorizes that the cat uses its four legs independently to effectively adjust its mechanical damping during impact [16]. This demonstrates how force plates are not only effective in measuring the magnitude of GRFs induced during impact, but also the intricate profiles and transient responses relevant to the design of mechanical and control systems in robots.

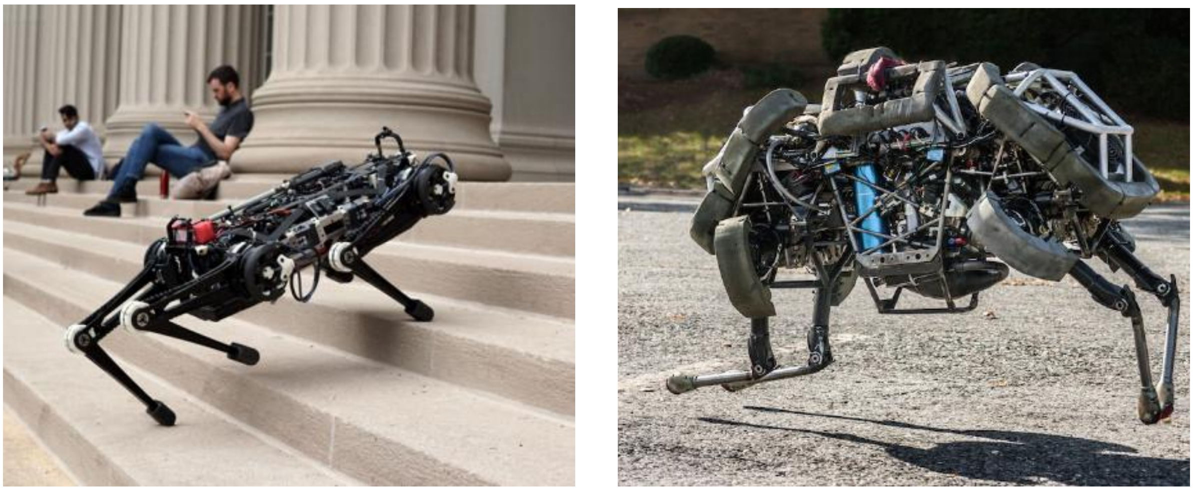
Liu *et. al.* [15] explored the use of GRFs in the developing the dynamic locomotion of their kangaroo-inspired biped robot. Force data acquired was used in analysis of the kangaroo's gait and provided insight into the synchronization of the animal's legs during its characteristic hopping motion. Dynamic centre-of-mass (CoM) trajectory models were consequently developed with the aid of this information [15]. This reveals another important use of force plate technology.

2.3.1 The Cheetah: an Inspiration for Robots

Arguably one of nature's most phenomenal athletes is the cheetah; a true master of speed and agility. Remarkable quadruped robots such as MIT's Cheetah 3 and Boston Dynamics' WildCat (the fastest quadruped robot in the world [14]) are testament to the attention that this incredible animal has attracted in the field of bio-inspired robotics at state-of-the-art institutions.

Figure 2.3: Image of a cheetah captured during high speed pursuit [61]

The interest in this amazing animal extends to UCT's own Mechatronics Laboratory with studies such as [17, 18, 19, 20] that have explored the integral role of the cheetah's tail in rapid dynamic manoeuvres. Current research in this laboratory is exploring the development of a fully-fledged quadruped cheetah robot and as explained above, detailed GRF profiles obtained from real cheetahs through the use of force plates would be of great use in the design and development of mechanical, control and trajectory planning systems, amongst others. This necessitates a further investigation into the dynamic motion of the cheetah as presented in Section 2.3.2 in order to guide the design specifications (such as sampling time and mechanical considerations) in this project.



(a) MIT Cheetah-3 [13]

(b) Boston Dynamics WildCat [14]

Figure 2.4: Images showing quadruped robots inspired by the cheetah

2.3.2 Dynamics of the Cheetah and other Quadruped Animals

2.3.3 Motion of the Cheetah

The cheetah is generally accepted as the world's fastest terrestrial animal (with a fastest recorded speed[1] of 29m.s^{-1} or 104.4 km.h^{-1} over a 200m track [22]) and is thus one of the most challenging yet exciting creatures to study.

Researchers from University of London's Royal Veterinary College conducted a detailed study comparing the kinetic and spatio-temporal characteristics of the cheetah and the racing greyhound (*Canis familiaris*) during high speed gallop. Although the racing greyhound gallops at a significantly lower top speed of 17m.s^{-1}, it shares very similar size, weight and gross morphology to the cheetah [22] and was thus chosen for comparison in the study. Hildebrand et al report that a cheetah running at 25m.s^{-1} completes a stride in only 0.28s with a stride length[2] of 7m [21].

Limb Forces During High Speed Gallop

The aforementioned study in [22] used an array of high-end Kistler force plates to measure GRFs experienced by the cheetahs and greyhounds during gallop. The experimental setup used in the study is presented in Figure 2.5 below:

In this study, 40 SLCs (single limb contacts where only one limb was in contact with a single plate) for the cheetah and 110 SLCs for the greyhound were collected. With increasing speed, both hind limbs experienced an increase in peak limb force but the vertical impulse experienced

[1] fastest speed recorded in a scientific study as of the time of writing of [22] (2011)

[2] stride length here is defined as the displacement between successive ground interactions of the same limb

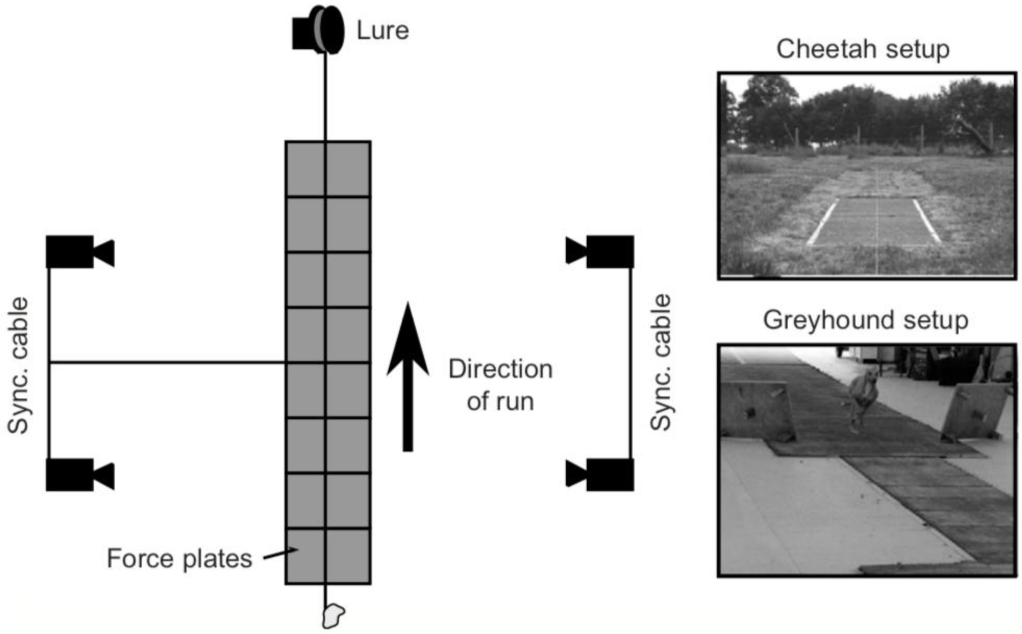


Figure 2.5: Diagram from [22] illustrating the experimental setup used in this study. It shows a series of force plates which the animals were enticed to run across by a mechanically-retracted lure. The setup also shows the high speed cameras used to capture footage for kinematic analysis.

remained unchanged [22]. It is evident that limb forces were increased to counteract decreasing stance (ground contact) time (as can be explained by Equation 2.2). Peak vertical hind limb forces experienced by the cheetah increased from 200 to 900N as speed increased from 8.7m.s^{-1} to 17.8m.s^{-1} (as measured in the experiment) [22]. Interestingly, the front limbs showed no correlation between peak vertical GRF and increasing speed with peak vertical GRF remaining in the region of 500 and 900N over the full speed range [22]. It was found in [22] that the cheetah supports 70% of its body weight on its rear limbs during gallop near 18m.s^{-1}. It is believed by the author that the particularly flexible spine of the cheetah allows it to position its rear limbs further forward and thus, more directly below its centre of mass [22]. This strategy would allow for more grip during acceleration (push off of the hind limbs). This demonstrates how, apart from its obvious quantitative value, detailed GRF data measured from force plates can be used by researchers to infer reasons for the behaviour and anatomical characteristics of animals during complex manoeuvres.

The vertical and anterior-posterior GRF profile for the cheetah and greyhound during gallop of >14m.s^{-1} is depicted in Figure 2.6 below. The curves of larger positive magnitude (for both the cheetah and greyhound) represent the vertical GRF component experienced. The mass and other physical characteristics of the cheetahs and greyhounds observed in this study are presented in Appendix D.1

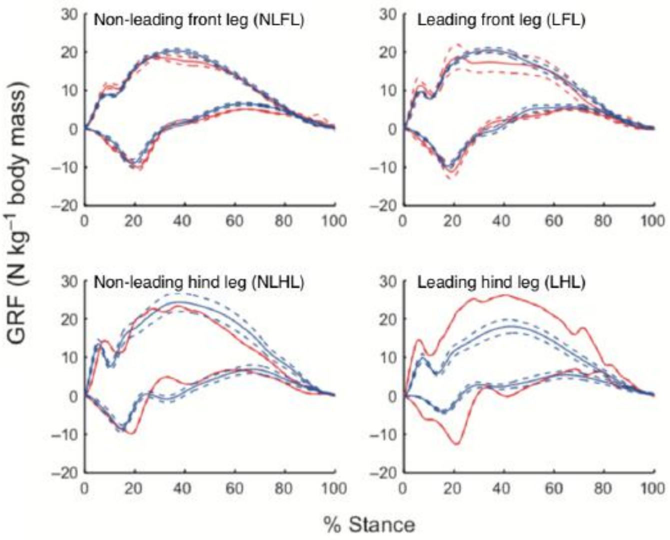


Figure 2.6: Figure from [22] modified by the author. Graphs show mean vertical and anterior-posterior GRF curves (with dotted ± standard error curves) for SLCs of each limb for cheetahs (red) and greyhounds (blue) at gallop speeds between 14m.s^{-1} and 18m.s^{-1}

Sampling Requirements

The current literature does not suggest any particular minimum sampling rate required in order to accurately capture the GRFs experienced by cheetahs during high speed manoeuvres. The study observing the motion of the cheetah and racing greyhound in [22] used force plates sampled at a frequency of 1kHz. For sports science and human gait analysis applications, Suchomel reported that several sampling rates ranging between 200Hz and 1kHz were used in various studies and that while some authors claimed that 200Hz is sufficient for posture and low speed gait analysis, others claimed that sampling rates of at least 500Hz were required to capture enough resolution (sufficient data points) to accurately reconstruct force-time curves for impulsive interactions [1]. Nyquist's sampling criterion states that, in order to prevent aliasing in a sampled signal, the minimum sampling rate of applied to the continuous signal is governed by:

$$f_s > 2f_{max} \tag{2.3}$$

where f_s is the sampling frequency and f_{max} is the maximum frequency in the signal.

2.4 Alternate mechanisms for GRF Estimation

Previous sections in this chapter have reviewed some of the *applications* of force plates. However, in order to validate the suitability of force plates in previously mentioned applications, an investigation into alternate mechanisms of estimating GRFs must be conducted.

2.4.1 Inertial Sensing

Various studies have investigated the use of MEMS IMUs in attempting to estimate 3-axis GRFs [23, 24]. Karatsidis *et. al.* were able to achieve RMS errors of 5.3%, 9.4% and 13.1% in estimating vertical, anterior and mediolateral GRFs respectively [24]. The study was conducted with 11 healthy male volunteers - each of whom wore a lycra suit with 17 IMU modules strategically placed in order to provide the required data for the kinematic model used. Having defined the kinematic relationships and intertial properties of all body segments in the model, the total external GRF, F_{ext}, was estimated using Newton's second law:

$$F_{ext} = \sum_{i=1}^{N} m_i(a_i - g) \tag{2.4}$$

where m_i and a_i were the mass and linear acceleration of the COM (*glossary*) of each model segment respectively and N was the number of total segments (16 in this study) [24]. The total external moment applied, M_{ext}, was also calculated using Euler's rotation equation for rigid bodies:

$$M_{ext} = \sum_{i=1}^{N}(J_i\dot{\omega} + \omega_i \times (J_i\omega_i)) - \sum_{i=1}^{N}\sum_{j=1}^{K_i}(r_{ij} \times F_{ij}) \tag{2.5}$$

where K_i was the number of end points in each segment (joints and contact points), ω_i and $\dot{\omega_i}$ were the angular velocities and accelerations of each segment respectively. [24] J_i represents the matrix of inertias around the COM of each segment, r_{ij} the position vectors between COM and end points of the segments and F_{ij} the resultant force in the end points of each segment [24]. This shows a feasible approach to estimating GRFs (and reaction moments) using direct computation instead of abstracting this to machine learning techniques. However, it should be emphasized that 17 IMUs are required to achieve the stated results and the system requires extensive calibration measures to allow for reliable and repeatable performance. Furthermore, the system relies on wearing of the suit to keep the inertial sensors at fixed, predefined points.

Similar studies performed in [23, 25] show promising results, however in each case, there are practical limitations that render intertial sensing as a means for accurate GRF estimation unfeasible for the objectives of this project. The need for many, precisely positioned IMUs and/or a specialized sensor suit is an example of limitations in [23] and [25]. In other studies, such as in [25], only the vertical GRF component could be reliably estimated using intertial sensing methods.

2.4.2 Discrete Pressure Mapping

Researchers at MIT investigated a novel force sensor design comprising of an array of barometric pressure sensors embedded in a flexible urethane rubber foot weighing less than 100g. The physical structure of the device is shown in Figure 2.7 below.

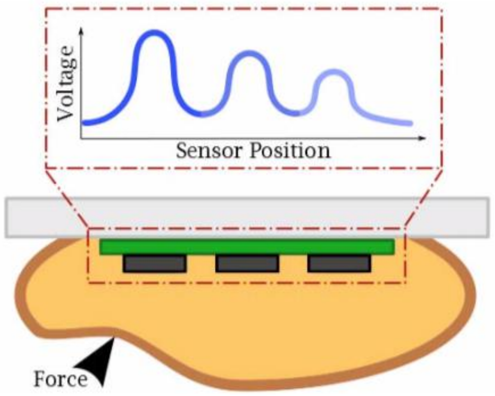


Figure 2.7: Diagram from [26] illustrating the working principle of the novel rubber force sensor designed in this study. A cross-sectional view of the sensor is shown with the urethane rubber being deformed by an applied force. The embedded array of barometric pressure sensors (black rectangles) mounted on the PCB substrate (green rectangle) is shown.

The composite construction exploits the predictable rubber deformation and the resulting pressure distribution in order to estimate 3-axis force measurements [26]. With the aid of a simple feed-forward neural network to map input pressure readings to 3-axis GRFs, the sensor was able to measure vertical GRFs up to 300N and a maximum of 80N in the two lateral axes. The experimental results indicating the performance of the sensor are presented in 2.1 [26]. While not commercially produced at this stage, the prototype sensor was produced for under $100 (USD) - a promising advancement in the pursuit of a more affordable 3 axis force sensor.

Table 2.1: Table to show experimental results indicating performance of the sensor designed in [26]

	Max	**RMSE**		**Max Error**	
	Force [N]	**[N]**	**[%FS]**	**[N]**	**[%FS]**
Fx	80	2.95	3.69	14.95	18.62
Fy	80	4.73	5.91	23.14	23.14
Fz	300	1.98	1.98	14.90	4.99

2.4.3 Capacitive Force/Torque Sensors

There are a number of commercially produced capacitive force/torque sensors that are currently available. These sensors typically come in small packages; a cylinder of diameter 80mm, height

25mm and weight 350g would constitute a typical size [30]. These sensors exhibit excellent sensitivity and linearity over their working range and several companies (such as ATI and Robotiq) produce sensors capable of sensing several thousand N of force with with sensitivity in the mN (milli newton) range.

Figure 2.8: Image of an industrial capacitive force/torque sensor (Axia80) made by ATI Industrial Automation from [30]

This technology typically utilizes two frames embedded in the sensor - one fixed and one movable. The frames are attached to a deformable component which is modelled as a spring and when forces/torques are applied to the sensor, deformation (strain) can be measured by capacitive elements. This can be understood by observing Equation 2.6 for the capacitance of a parallel plate capacitor below:

$$C = \frac{\epsilon A}{d} \tag{2.6}$$

where C is the capacitance of the parallel plate capacitor, ϵ is the absolute dielectric constant, A is the area between the capacitor plates and d is the displacement between the plates. Small deformations as mentioned above lead to varying displacement d between the capacitor plates and can be measured as a change in capacitance.

While these sensors perform particularly well in almost all performance characteristics, their drawback is their exorbitant cost. They are typically intended for industrial automation and robotics applications where high precision, reliability and mobility are required.

Several studies [27, 28] have explored the production of lower cost multi-axial GRF sensors and have managed to achieve particularly promising results; sensor designs with comparable accuracy and sensitivity to industrial models (such as the ATI sensors mentioned aboved) have been created and tested in these studies. However, the sensors developed in [27, 28] (and several others) have the limitation of the range of force that can be sensed - typically a maximum normal (vertical) GRF force in the order of 10 newtons is detecable and thus, these technologies are not suitable for this project where forces in the region of several hundred newtons are expected.

2.5 Force Plates

Force plates are widely considered the gold standard in measuring GRFs for a wide range of applications due to their excellent accuracy and repeatability. While most internal designs of commercially-sold force plates are proprietary and kept secret, common sensing technologies that form the basis of transducers in commercial force plates are explored below.

2.5.1 Sensing Mechanisms in Force Plates

Hall effect

In Hall effect variants, multiple Hall effect sensors and magnets are arranged within the force transducer to measure changes in the magnetic field experienced caused by deflecting spring elements under exertion of externally applied forces and moments [31]. The draw back for this simpler design is that only lower sampling rates (compared to strain gauge technology) are attainable and plates with this technology typically exhibit lower stiffness, resulting in a lower natural frequency of the platform and more risk of resonant vibrations [31].

Piezoelectric

Piezoelectric force transducers rely on piezoelectric crystals, such as quartz, at the core of the sensor. When subjected to an external force, piezoelectric materials produce an electric charge proportional to the mechanical stress exerted on them. This property is exploited to determine applied forces and moments by arranging separate piezoelectric materials in configurations such that they are sensitive to stress in particular directions. A particularly desirable quality of piezo sensing elements used in force plates is that they cover very large measuring ranges and can be made to be only a few millimetres thick [33]. Furthermore, due to their exceptionally high rigidity, piezo sensors exhibit negligible deformation under operational loading conditions and are thus can be flexibly integrated into a variety of structures. Unfortunately, however, piezo transducers are particularly prone to charge drift over time (up to as much as 10N per minute [33]) and require frequent zeroing or high pass filtering to suppress drift.

Strain Gauges

Strain gauge technology is most popular in high-end strain gauges and provides the most accurate and flexible force measurement [31]. In order to understand how strain gauge sensing works, the equation for the resistance of a material with certain geometry is defined below:

$$R = \frac{\rho l}{A} \tag{2.7}$$

where ρ represents the resistivity (a property of the material), l the length of the material and A its cross sectional area. Strain gauges are resistive transducers that vary their resistance in response to an applied mechanical strain, where strain $\epsilon = \frac{\Delta l}{l}$ represents the ratio of the deformed length of a material to its original length in response to mechanical stress. The most widely used strain gauge structure is the bonded metallic pattern as depicted in Figure 2.9 below. The metallic foil grid pattern as seen in Figure 2.9 above maximizes the amount of resistive material subject to strain in the direction parallel to its longitudinal axis. The metallic grid is bonded to a backing material called the carrier which is mounted directly onto the test specimen. This allows strain experienced by the test specimen to be transferred directly to the gauge which responds with a linear (proportional) change in resistance [35]. Another important parameter of a strain gauge is its sensitivity to strain which is quantified by the gauge factor S. This is defined by equation 2.5.1 and is the ratio of the fractional change in resistance to the fractional change in length (strain) [35]:

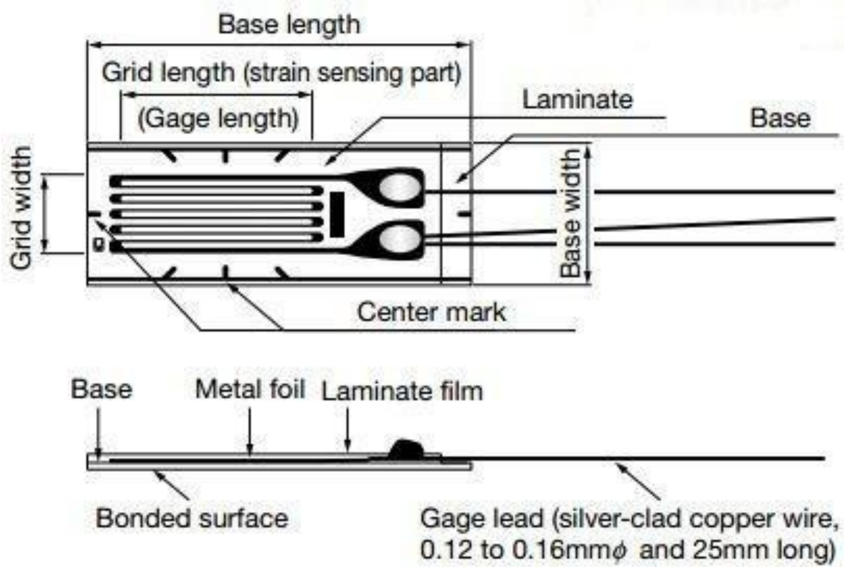


Figure 2.9: Diagram of a bonded metallic strain gauge from [34]

$$S = \frac{\Delta R/R}{\Delta l/l} = \frac{\Delta R/R}{\epsilon} \tag{2.8}$$

where R and l are as previously defined. Strain gauges are typically connected in a Wheatstone bridge configuration depicted by the circuit shown in Figure 2.10. A full Wheatstone bridge comprises of four strain gauge elements and allows for the maximum differential voltage change in response to changing resistance of the gauges. Additionally, it allows for good immunity to temperature drift as all strain gauges will react almost identically to changes in temperature and will thus experience very little *relative* changes in their resistance (provided they are identical in structure and are exposed to the same source of heat which is near uniform with respect to their geometry).

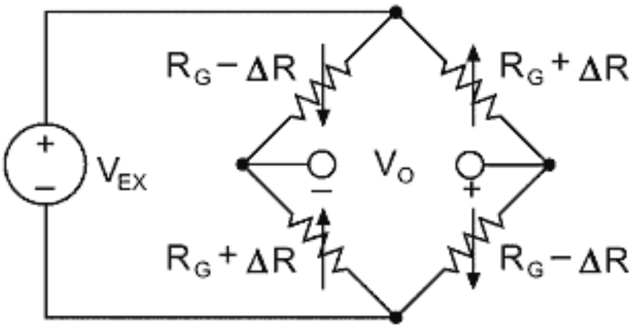


Figure 2.10: Circuit diagram from [35] of a full Wheatstone bridge with four sensing elements. V_{ex} represents the bridge excitation voltage and V_o the output differential voltage.

2.5.2 Current State-of-the-Art

The BP400600 force platform made by AMTI, a prominent force plate manufacturer, is considered to be one representative of the current state-of-the-art based on its specifications/performance [31]. It uses many thin-walled cylindrical sensing elements, each of which are coupled with strain gauges to isolate signals produced by individual forces and moments - allowing them to be measured independently [31]. Furthermore, signal conditioning - including amplification and filtering - most commonly forms part of the product and hence, is performed within the plate structure so as to allow for processed signals to be measured directly by a computer or digital serial interface. To illustrate the performance of a typical industrial grade force plate, relevant specifications of the AMTI BP400600 force platform are shown in 2.2 below:

Table 2.2: Relevant performance specifications of the AMTI BP400600 force plaform

Parameter	**Value**
Fx, Fy capacity [N]	2224
Fz capacity [N]	4448
Fx, Fy sensitivity [μV/(V*N)]	0.67
Fz sensitivity [μV/(V*N)]	0.75
Measurement accuracy [% of load, all channels]	±0.25
Crosstalk [%, all channels]	<2
Non-linearity [% full scale, all channels]	±0.2
Hysteresis [% full scale, all channels]	±0.2

Furthermore, the platform measures 400 x 600 x 82.55 mm (width x length x height) and weighs 31.82kg. A picture of the platform is show in 2.11 below.

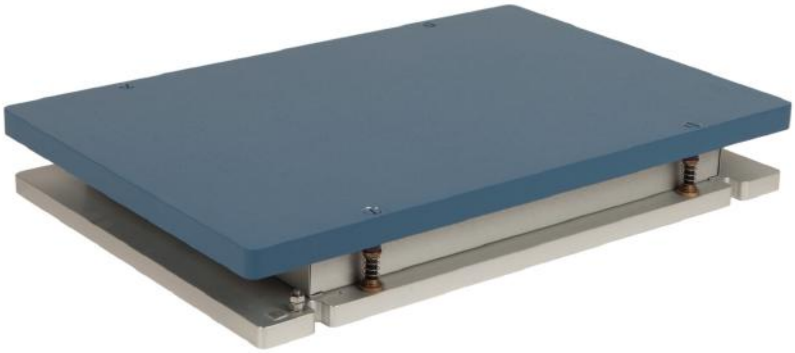


Figure 2.11: Image of the BP400600 force platform made by AMTI [31]

2.5.3 Force Plate Pricing

While sophisticated sensing technology as mentioned in 2.5 above allows for very accurate sensing and low cross-talk, industry-standard force plates such as the BP400600 carry an extremely high cost. Considering that a core criterion of this project is the cost of the developed system, due consideration must be given to the cost of existing force plate systems to form a reference price range.

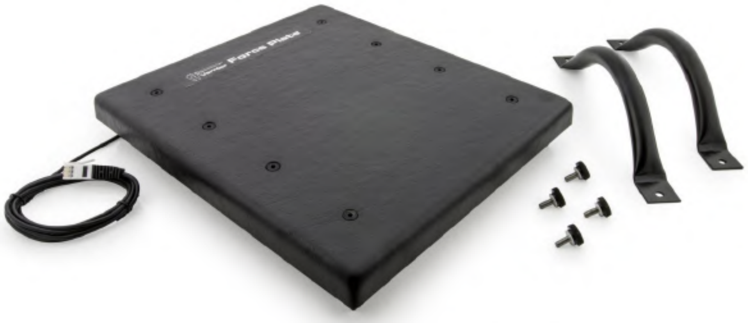


Figure 2.12: Image of the Vernier Force Plate [32]

One particular low cost platform is the Vernier Force Plate - a product intended for experimentation in schools and other teaching environments. This product is only able to measure the magnitude of the normal/vertical component of the GRF vector (and not the individual 3 axis force vectors) to a maximum of 3500N in compression and 850N in tension [32]. This product is priced at $374 (USD) [32] - equivalent to R5236 assuming an exchange rate of R14/$.

Force plates manufactured by companies such as AMTI and Kistler (such as the strain-gauged-based AMTI BP400600) represent the gold standard in current commercial force plate technology and these models are typically priced in the region of $30 000 - $40 000 (USD) [36]. Cheaper models made by manufacturers such as ACME International are priced in the range of $8000 (USD) [36]. A quotation (found in Appendix D.1) obtained from HiTech Therapy CC (a

South African company selling rehabilitation and exercise products) quotes the price of the AMTI BP400600 at R300 000 (as of the date of writing this report). A slightly larger AMTI model (1200x1200 mm top plate area), the BP12001200, with almost identical performance specifications, was listed at R940 000 in this same quotation.

2.6 Summary of Literature

This review of relevant literature has explored the various applications of force plates as currently used in practice. The use of force plates in biomechanical analysis of human motion and their utility in athletics and sports science rehabilitation applications has been noted. Furthermore, the use of force platforms in the clinical diagnosis of human disorders causing deficient motor function such as Parkinson's Disease and MS has been explored with reference to studies in [12, 9, 10]. From these findings, it is evident that force plates perform an integral role in the study of human biomechanics and their practical use in sports science and clinical pathology is extensive.

Biomimicry has proven to be an exciting driver for innovation in modern robotics. Robotic imitations of some of nature's most impressive athletes such as the MIT Cheetah 3, as outlined in [13], are not only impressive academic achievements but also present exciting prospects for a number of practical applications such as search and rescue operations. They are indicative of what can be learnt from nature and millions of years of refinement in natural technology through evolution.

Previous studies performed on the dynamics the cheetah's motion [21, 22] provide great insight into kinetic and spatio-temporal characteristics of these animals during high speed gallop. Findings showed that peak normal (vertical) GRFs of approximately 900N were recorded by the cheetahs in [22] and peak anterior-posterior shear forces of approximately 150N according to findings in the same study. It is worth noting that the *vertical* GRF profile, as well as the *vertical* impulse and other derived quantities, were of main focus in [22, 21] but the anterior-posterior GRF component is also observed during brake and take off of the limb in contact with the ground. Interestingly, these studies do not consider the third (mediolateral) GRF component in analyzing the cheetah's motion despite the fact that full 3 axis force platforms were used. It is presumed that this is due to the fact that this component was of negligible magnitude relative to the other two GRF components or did not provide any meaningful information in the experiments conducted in [22, 21].

Apart from force plates, several techniques for sensing and measuring GRFs have been reviewed. While most of these alternate techniques achieve sufficient accuracy, many have shown to be impractical for use in the applications outlined in sections 2.2 and 2.3.1. Sensing using inertial measurements as in [24] required 17 strategically placed IMUs and would be impractical as a means of gathering GRF data even on captively-bred cheetahs, let alone those in the wild. Even if they could be placed accurately on the animals which would likely require sedation, the animals would, in all likelihood, not run naturally as they would when unhampered by foreign devices placed on them. Similarly, capacitive force sensors traditionally used in industrial automation applications such as the ATI Axia80 (Figure 2.8) are fully capable of accurately measuring 3 axis GRFs and moments, however they are too large and bulky to place directly on animals or human subjects without hampering motion as described previously. Furthermore, these sensors are extremely costly [36].

While a review of the state-of-the-art in force plate technology revealed that many extremely high precision platforms exist and present a viable mechanism for measuring GRFs in many applications, a review of the prices of these industrial grade plates conducted in Section 2.5.3 revealed how exorbitantly expensive these devices currently are. In conclusion, it is evident that there exists a need for a robust, accurate and practical low cost force plate capable of measuring 3-axis GRFs.

Chapter 3

Theory Development

3.1 Machine Learning for Regression

Machine learning techniques such as neural networks and other mechanisms explored in subsequent sections below present an extremely powerful set of tools in regression problems such as the challenge faced in this project. The real power of this approach lies in the fact that these techniques require little to no knowledge of internal system dynamics or an explicitly defined input-output transfer function.

Consider a dataset with N observations, each with an associated response or target y_i. In general, each observation will be a vector of p features (or predictors, in the case of a testing or query observation), denoted $\mathbf{x_i} = [x_{i1}, x_{i2}, ..., x_{ip}]^T$ for observation i. The target for each observation may be a scalar, y_i, or a vector $\mathbf{y_i} = [y_{i1}, y_{i2}, ..., y_{iq}]^T$ in the case of multi-value regression (or classification) problems with q outputs. The objective is generally to model the relationship between predictors and targets so as to most *accurately*[1] predict the system's response for a given set of features [41]. This approach represents a black box model of the system - it is assumed that the system can plausibly be defined by an underlying mathematical structure but the objective is not to explicitly uncover this structure but rather to encode it in a model or network using only a considerable number of observations and their corresponding responses. The aim is thus to rather design a model or *hypothesis*, as defined in the language of modern analytics [41], to discern patterns in the data collected that will allow for prediction of future responses of the system when target responses are no longer available.

[1] Accurately is used with caution here as the notion of accuracy can be specific to the problem at hand. Typically, though, in regression problems, accuracy refers to how closely predicted system responses match their actual counterparts. Quantitative measures of this are explored in subsequent sections.

3.1.1 Linear Regression and General Linear Machines

Linear regression models assume linear relationships between the response variable y and the p dimensional vector of regressors $\mathbf{x}^T = [x_1, ..., x_p]$. The regressors are strategically chosen (linearly independent) basis functions that characterise the type of regression. To better illustrate this, consider the following equation

$$y_i = \beta_0 1 + \beta x_{i1} + ... + \beta x_{ip} + \epsilon_i = \mathbf{x}_i^T \beta + \epsilon_i \tag{3.1}$$

This may be expressed in matrix notation by the following equation

$$\mathbf{y} = \mathbf{X}\beta + \epsilon \tag{3.2}$$

where

- $\mathbf{y}$ is the response vector containing observations of the dependent variable [37]. This is not to be confused with the vector of predicted outputs, typically denoted $\hat{\mathbf{y}}$.
- $\mathbf{X}$, typically termed the 'design matrix', is a matrix of p-dimensional row vectors (corresponding to each observation) containing basis functions called regressors [37]. Most commonly, the first regressor in each row is a constant ($x_{i0} = 1$ for $i = 1, ..., n$ where n is the number of observations). This is to allow for a non-zero intercept (β_{i0}) in the linear model [37]. Often, non-linear basis functions (regressors) are selected such as n^{th} order polynomial terms or Gaussian kernels. Regardless of independent basis functions, the model remains linear if it is linear in the vector β [37].
- β is a $(p+1)$ dimensional vector (or p dimensional if the constant first regressor is included in the model) containing the regression co-efficients.
- ϵ is a column vector containing error terms ϵ_i for each observation. This parameters contains all other influences on the response variable $\mathbf{y}$, other than that of the regressors. The objective is to minimise this error term for each observation while still maintaining sufficient generalisation to other data sets.

3.1.2 The Support Vector Machine (SVM)

Due to the detail and mathematical complexity required to thoroughly elucidate the SVM algorithm and its associated structures, as well as the fact that this model is not used extensively in this project, only a brief introduction to this topic is presented in this report.

Considering a training data with N observations as detailed in Section 3.1. A particularly popular method introduced by Vapnik in 1995 [38] is known as ϵ-SV regression (support vector regression using an *epsilon-insensitive* cost function). This SVM strategy is explored in this report as the MATLAB implementation used in this project leveraged ϵ-SV regression.

This method seeks to produce a function $f(x)$ whose estimated outputs deviate from the targets (true responses) y_i by at most ϵ for each observation i [39]. The notion of the predictor being **epsilon-insensitive** means that it will treat all errors within the range $y_i \leq |\epsilon|$ as zero. Following the notation presented in [39], consider a linear function for $f(x)$

$$f(x) = \mathbf{w} \cdot \mathbf{x} + b \tag{3.3}$$

where $\mathbf{w}$ and b are analogous to β and ϵ in Section 3.1.1 respectively. The first objective is to obtain a maximally *flat*[2] predicted response which can achieved by minimising the norm of the weights vector w [39]. This resolves to the following convex optimisation problem:

$$\begin{aligned} &\text{minimise } \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to } \begin{cases} y_i - \beta \cdot x_i - b \leq \epsilon \\ \beta \cdot x_i + b - y_i \leq \epsilon \end{cases} \end{aligned} \tag{3.4}$$

This implies that the optimisation problem is feasible and that a function f actually exists that can produce estimated responses to within a precision of ϵ for *each* observation [39]. Practically, this is often not the case and consequently, the idea of *slack variables* ζ_i and ζ_i^* is introduced to deal with otherwise unfeasible constraints. The formulation presented in equation 3.4 above is thus adapted to

$$\begin{aligned} &\text{minimise } \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{p}(\zeta_i + \zeta_i^*) \\ &\text{subject to } \begin{cases} y_i - \beta \cdot x_i - b \leq \epsilon + \zeta_i \\ \beta \cdot x_i + b - y_i \leq \epsilon + \zeta_i^* \\ \zeta_i, \zeta_i^* \geq 0 \end{cases} \end{aligned} \tag{3.5}$$

The constant, positive parameter C controls the balance between tolerance of prediction-target deviations greater than ϵ and the flatness of the estimated response of f. This is termed the soft margin loss setting and is clarified by Figure 3.1 below

As seen above, only prediction deviations beyond the shaded ϵ region are penalised linearly (proportional to the magnitude of their deviation beyond this region).

[2] a response that captures the behaviour of the response variable but with no oscillations or high frequency errors

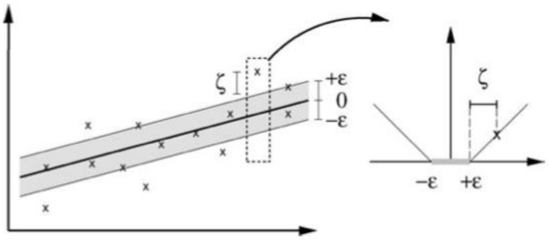


Figure 3.1: Visual representation of the soft margin loss principle for a linear $\epsilon - SV$ regressor with slack variables ζ_i and ζ_i^* [39]

3.2 Neural Networks

3.2.1 The concept of an artificial neuron

In order to understand the structure of a feed-forward neural network, it is imperative to first gain a greater understanding of possibly the most fundamental unit; the artificial neuron. The most basic type of artificial neuron is the perceptron - a model first developed in the 1950s and 1960s by Frank Rosenblatt [42]. Perceptrons take several inputs $\mathbf{x_i} = [x_{i1}, x_{i2}, ...]^T$ to produce a single *binary* output y as indicated in Figure 3.2 below:

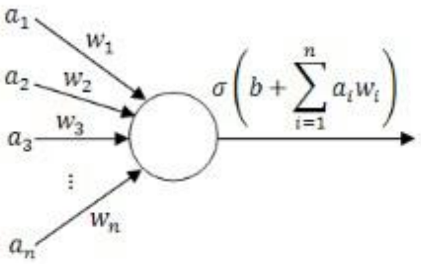


Figure 3.2: Diagram to show the basic structure of Rosenblatt's perceptron - the most basic artificial neuron [42]

Rosenblatt proposed the simple concept of **weights** $[w_1, w_2, ...]$ to compute the perceptron output. These weights are merely real numbers that quantify the importance or scaled contribution of individual inputs in computing the output. The output y of the perceptron is binary - either 0 or 1 - and is determined by comparing the weighted sum of the perceptron's inputs to a predefined **threshold value**, θ, that forms another parameter of the perceptron. This formula to compute the perceptron output y is captured by equation 3.6 below:

$$y = \begin{cases} 0 & \sum_j w_j x_j \leq \theta \\ 1 & \sum_j w_j x_j > \theta \end{cases} \tag{3.6}$$

This relationship can be more elegantly expressed by writing the weighted sum $\sum_j w_j x_j$ as the dot product of input and weight vectors respectively, i.e. $\mathbf{w} \cdot \mathbf{x} = \sum_j w_j x_j$. Furthermore, it is typical capture the perceptron's decision threshold θ as a **bias** b of the perceptron, where $b = -\theta$. Equation 3.6 can now be expressed as:

$$y = \begin{cases} 0 & \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ 1 & \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases} \tag{3.7}$$

It should be clarified that each perceptron has a vector of weights $\mathbf{w} = w_1, w_2, ..., w_q$ corresponding to its q inputs $\mathbf{x} = x_1, x_2, ..., x_q$ but only *one* scalar-valued bias b. The bias can be understood as a tuning parameter that controls how easily a neuron will activate (produce and output of 1) or in biological terms, how readily that neuron will *fire* [42].

3.2.2 Activation functions

The perceptron model explored above was only capable of producing binary outputs. In trying to tune the network weights and biases to fit the application at hand, suppose a small change ΔW in the weights of a particular perceptron was made. Ideally, the output y of the perceptron (and consequently, the network) should change by some small amount Δy as depicted in Figure 3.3. However, this is not the case, as even a small increment in these parameters could cause a perceptron to flip its output from 0 to 1 (or *vice versa*). Furthermore, the obvious limitation exists that this model cannot produce continuous outputs as would be required for regression problems.

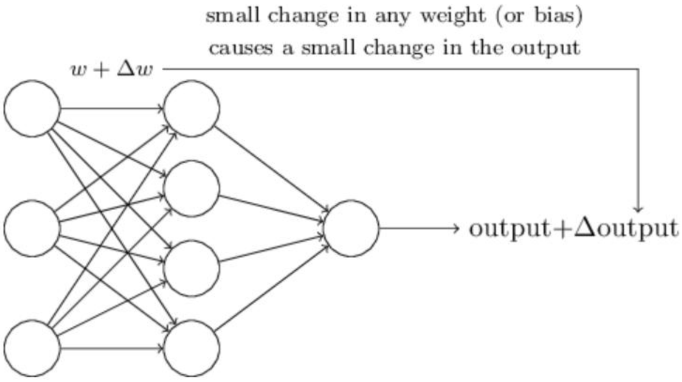

small change in any weight (or bias)
causes a small change in the output
$w + \Delta w$
output+Δoutput

Figure 3.3: Diagram from [42] to show the desired network behaviour; tuning the network weights by a small increment Δw produces an comparably small increment in the computed output Δy

This leads to the introduction of the sigmoid neuron and activation functions. The sigmoid neuron follows the exact same principle as the perceptron, however, it accepts both continuous inputs for $\mathbf{x} = x_1, x_2, ..., x_q$ and produces a continuous (non-binary) output by applying the sigmoid function, $\sigma(z) = \frac{1}{1+e^{-z}}$, to the perceptron output. As such, the the output of the sigmoid neuron becomes

$$y = \sigma(\mathbf{w} \cdot \mathbf{x} + b) \tag{3.8}$$

More generally, the activation function need not be the sigmoid function. Instead, several common functions exist, as outlined in Table 3.1 below [41]. Graphs of these activation functions and their gradients are plotted in Figure 3.4.

Table 3.1: Table to show commonly used activation functions and their algebraic formulae

Activation Function $\sigma(z)$	**Expression**
Linear	$\sigma(z) = z$
Sigmoid	$\sigma(z) = \frac{1}{1+e^{-z}}$
Hyperbolic tangent	$\sigma(z) = \tanh(z) = \frac{e^{2z}-1}{e^{2z}+1}$
ReLU	$\sigma(z) = \max(0, z)$

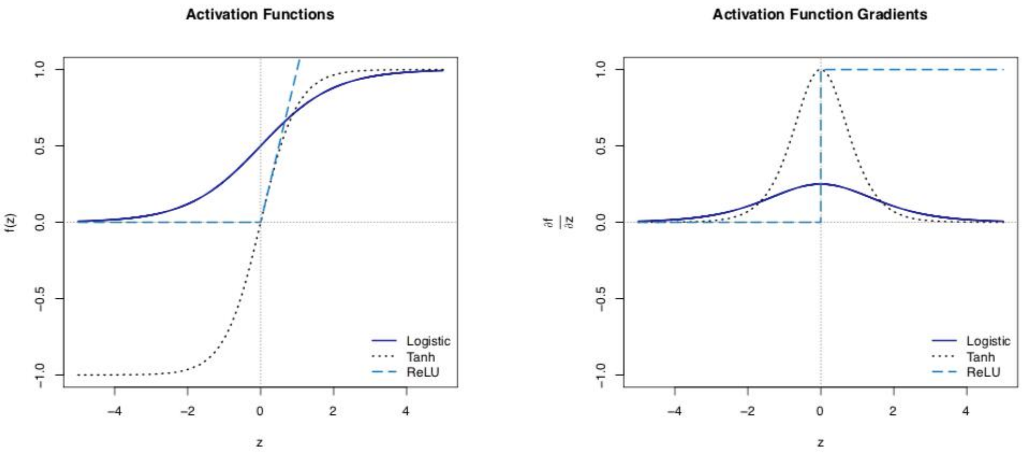


Figure 3.4: Diagram from [41] to provide a graphical representation of commonly used activation functions and their gradients.

The linear activation function $\sigma(z) = z$ is particularly useful in the output layer of regression networks where a continuous numerical output is needed. This is in contrast to classification problems where sigmoid and hyperbolic tangent functions are more often used so that the output behaves more like a probabilistic distribution [41]. However, non-linear activations are commonly used in hidden layers of both regression and classification problems. The activation function gradients observed in 3.4 above reveals that for very large inputs, there is very little change in value of the activation function [41] - closer to the desired behaviour described in Figure 3.3 above.

3.2.3 Network architecture

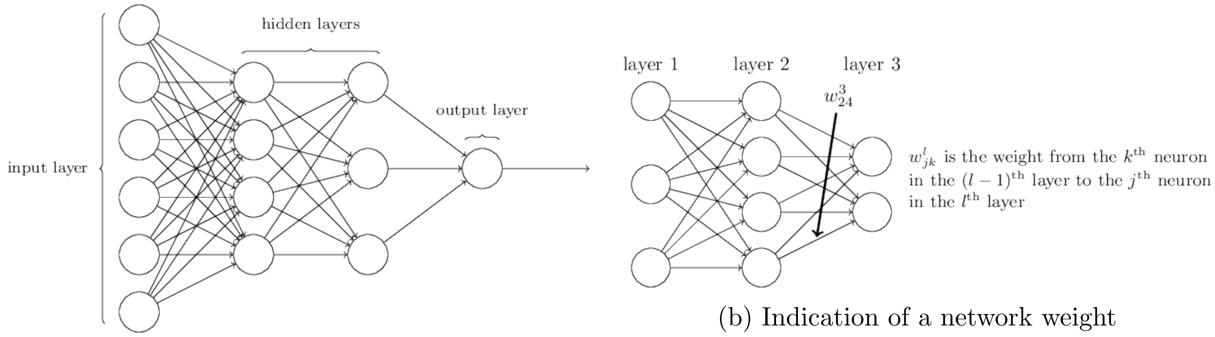
Following the notation used in [42], let a_j^l signify the jth neuron in the lth layer of a regular feed-forward neural network with $J-1$ hidden layers. The network structure can be expressed by the feed-forward updating equation in 3.9 below:

$$a_j^l = \sigma_l(\sum_{k=1}^{d_{l-1}} a_j^{l-1} w_{kj}^l + b_j^l) \text{ where } l = 1, 2, ..., J\,; j = 1, 2, ..., d_l \tag{3.9}$$

where:

- $\sigma_l()$ is the activation function applied to layer l
- d_{l-1} is the number of neurons in layer $l-1$ (and d_l the number of neurons in layer l)
- w_{kj}^l denotes the weight parameter connecting the kth neuron in layer $l-1$ and node j in layer l
- b_j^l is bias corresponding to the jth neuron in layer l.

The feed-forward equation is evaluated with the input vector $\mathbf{x_i} = [x_{i1}, x_{i2}, ..., x_{ip}]^T$ as values associated to the first layer of neurons - the input layer in Figure 3.5 below. Algebraically, this is expressed as $a_j^{(0)} = x_{ij}$ where index i denotes the the ith observation and index j denotes the jth data feature or input neuron.


hidden layers
input layer
output layer
layer 1
layer 2
layer 3
w_{24}^3
w_{jk}^l is the weight from the k^{th} neuron in the $(l-1)^{\text{th}}$ layer to the j^{th} neuron in the l^{th} layer

(a) Basic network structure [42]

(b) Indication of a network weight

Figure 3.5: Diagram from [42] to show the basic architecture of a feed-forward neural network

3.2.4 Cost Functions and Learning

In order to train neural networks by tuning the many weights and biases that characterise the relationships encoded in the network, it is crucial to define a mechanism to quantify the predictive accuracy of the network - the ‘quality’ of the fit. In regression problems, a prediction under a particular network configuration maps the inputs presented to the network to a vector in the response vector space [41] (in the case of multivariate outputs). A natural measure of the quality of the fit for each observation is the squared distance between the prediction and actual response vectors for a particular observation [41]. This is quantified by the quadratic **mean-squared-error** (MSE) cost function as defined below:

$$C_{MSE}(\mathbf{w},\mathbf{b}) = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{y}_i - \mathbf{\hat{y}_i})^2 \tag{3.10}$$

where N is the number of observations, $\mathbf{y_i}$ is the vector of actual responses and $\mathbf{\hat{y}_i}$ is the vector of predicted responses for the ith observation. The central objective of fine tuning weights and biases of the network through training is to minimise the cost function $C(\mathbf{w},\mathbf{b})$ [42]. Other cost functions that are more suited to classification problems, such as the cross-entropy function, are not explored in this project.

Gradient Descent

Gradient descent (and its derived methods explored in subsequent sections) is a simple but powerful training algorithm used in neural networks to minimise a cost function. It forms the basis of more powerful learning techniques such as the Levenberg-Marquardt algorithm which combines gradient descent with the Gauss-Newton numerical method. In order to more easily understand and visualise the strategy used by this algorithm, it is useful to consider an arbitrary cost function $C(\mathbf{v})$ of two variables where input vector $\mathbf{v} = [v_1, v_2]$. This allows the cost function to be visualised as a surface, as depicted in Figure 3.6 below.

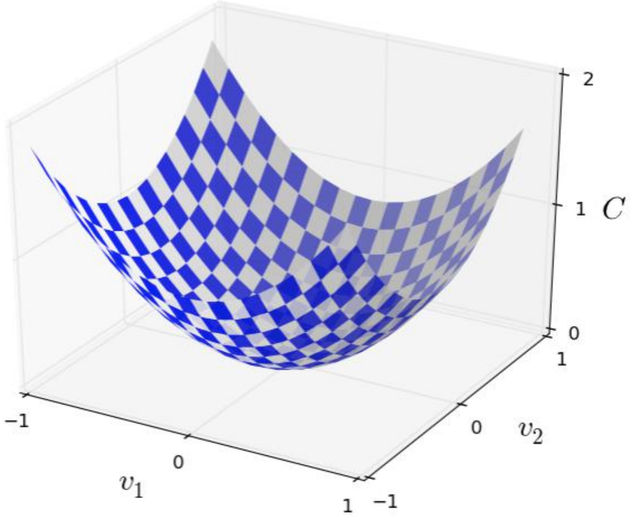


Figure 3.6: Visual representation of a theoretical cost function surface $C(\mathbf{v})$ of two variables [41]

Finding a global minimum of $C(v_1, v_2)$ analytically through calculus seems a plausible strategy when considering a smooth and well-behaved surface of two variables as in Figure 3.6 above. However, with even a modestly sized neural network with anywhere between thousands and millions of weight and bias parameters, computing gradients of such a cost function (a function of all these weights and biases) and finding a global minimum directly using calculus becomes completely intractable.

The gradient descent algorithm does use multi-dimensional gradients but does not attempt to solve for stationary points directly. Consider an incremental change ΔC to the cost function:

$$\Delta C \approx \frac{\partial C}{\partial v_1}\Delta v_1 + \frac{\partial C}{\partial v_2}\Delta v_2 = \nabla C \cdot \Delta \mathbf{v} \tag{3.11}$$

where $\nabla C = (\frac{\partial C}{\partial v_1}, \frac{\partial C}{\partial v_2})^T$ represents the gradient vector of cost function C. In order to reduce the cost function, ΔC must clearly be negative. The gradient vector (or more specifically, the negative thereof) determines he direction of $\Delta \mathbf{v}$ that will yield the largest rate of decrease in the cost function. Choosing

$$\Delta \mathbf{v} = \eta \nabla C \tag{3.12}$$

ensures this. A small (typically $\ll 1$), positive parameter η, often termed the *learning rate* [42], is introduced to limit the step size of each adjustment of the network parameters. An optimal learning rate is often chosen empirically [3]. This guarantees that the expression for ΔC in equation 3.11 is always negative and provides a method of following the gradient to a minimum by consistently applying the update formula [42]

$$\mathbf{v} \longrightarrow \mathbf{v}' = \mathbf{v} - \eta \nabla C \tag{3.13}$$

3.2.5 Bias-variance Trade-off and Regularisation

An important consideration in the design of a regression model is its complexity and embedded in this, its predictive power. However, more important is the model's ability to capture underlying patterns in the training data presented to it and consequently, its capacity to generalise to out-of-sample (unseen) data [41]. If a model is too complex, it will likely over fit (recall) the training data presented to it (right plot of Figure 3.7) and will suffer from high variance across unseen data sets [41]. If a model is too simple, it will not be able to capture underlying patterns and nuances in the data and will under fit the data due to a large bias (left plot of Figure 3.7).

Regularisation is used to constrain the complexity of the model and provide greater robustness and generalisation performance. One way to achieve this is to penalise the complexity of the model. Consider including the following constraint on the network weights

$$\sum_{j,k,l} (w^l_{kj})^2 \leq \vartheta \tag{3.14}$$

for some non-negative scalar ϑ where increasing ϑ produces a more constrained model. This leads to a penalised cost function as defined below:

[3] choosing a larger learning rate will result in faster training (minimisation of the cost function) but also increased risk of missing a global minimum in the cost function due to overshoot and less exploration of the cost function domain

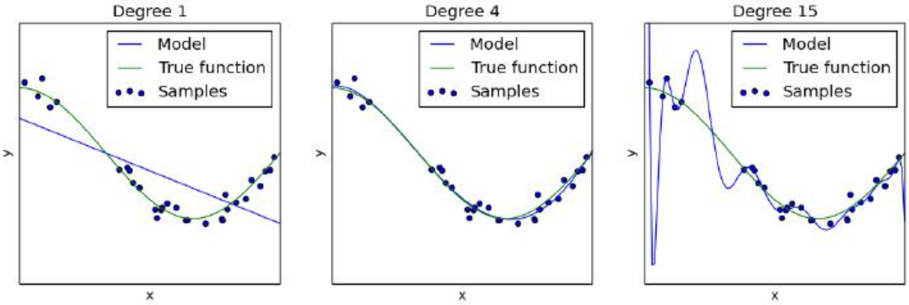


Figure 3.7: Visual representation showing predictions (variable y) of regression models of varying complexity (nth order polynomial terms for increasing n) with input variable x [43]. The middle plot is typical of a model with a good balance of complexity and robustness (generalisation ability).

$$C^*_{MSE}(\mathbf{w},\mathbf{b}) = \sum_{i=1}^{N}(\mathbf{y}_i - a_1^L)^2 + \lambda \sum_{j,k,l}(w^l_{kj})^2 \tag{3.15}$$

where λ corresponds directly to ϑ: increasing λ serves to constrain and simplify the model and decreasing it has the opposite result.

3.3 The Extreme Learning Machine (ELM)

The ELM, first proposed by Huang [45], has become an popular area of research in the field of computational intelligence. The ELM proposes an alternative to traditional SLFNs (single-hidden-layer feed-forward networks) in regression and multi-class classification problems with the significant and unique advantage that it does not require tuning of its hidden layer [45]. This presents a significant advantage; gradient descent-based algorithms used in many traditional learning mechanisms are often very slow and may converge to local minima [46] in minimisation of the cost function. The performance of of a regularised ELM with a sufficient number of hidden neurons[4] is competitive with standard approaches such as neural networks leveraging back propagation or the SVM and variations thereof [45][40]. However, the ELM is typically hundreds to thousands of times faster to train than these comparable methods [40].

3.3.1 Structure

The basic structure of a SLFN ELM network is outlined in 3.8 below. An approach unique to the ELM is its randomly selected weights and biases in its hidden layer [46]. As such, these parameters need not be adjusted through algorithms like back-propagation (BP), but instead, only the output weights are tuned through linear regression and pseudo-inversion [40] as explored below.

[4] sufficient is clearly a relative and application-specific term. However, it is shown in subsequent sections how empirically determining and optimal number of hidden neurons is a simple task due to the efficiency of the ELM.

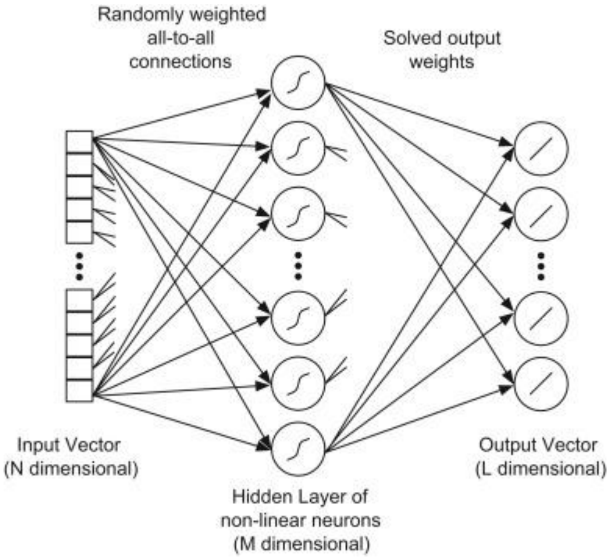


Figure 3.8: Diagram from [44] depicting the basic structure of a single hidden layer ELM. Only the output weights are solved through linear regression (using pseudo-inversion)

For N distinct observations, $\hat{N}$ hidden neurons and m target parameters in the target/response matrix $\mathbf{T}$

$$\mathbf{H}\beta = \mathbf{T} \tag{3.16}$$

where

$$\mathbf{H} = \begin{bmatrix} \sigma(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \dots & \sigma(\mathbf{w}_{\hat{N}} \cdot \mathbf{x}_1 + b_{\hat{N}}) \\ \vdots & \dots & \vdots \\ \sigma(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \dots & \sigma(\mathbf{w}_{\hat{N}} \cdot \mathbf{x}_{\hat{N}} + b_{\hat{N}} \end{bmatrix}, \beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\hat{N}}^T \end{bmatrix}_{\hat{N}\times m} \text{ and } \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N\times m} \tag{3.17}$$

$\mathbf{H}$ is typically termed the 'hidden layer output matrix' [46] or 'design matrix' since column i of $\mathbf{H}$ corresponds to the to the ith hidden neuron output with respect to the inputs $x_1, x_2, ..., x_N$ [46].

3.3.2 Training through pseudo-inversion

Training of the ELM simply reduces to solving for the $\beta_{[\hat{N}\times m]}$ matrix of output weights. For fixed, random input weights $\mathbf{w}_i$ and biases b_i, this task is equivalent to finding the least squares solution $\hat{\beta}$ for the linear system defined in 3.3.1:

$$\|\mathbf{H}\hat{\beta} - \mathbf{T}\| = \min_{\beta} \|\mathbf{H}\hat{\beta} - \mathbf{T}\| \tag{3.18}$$

It follows that if $N = \hat{N}$; that is, the number of hidden neurons is equal to the number of training observations; **H** will become a square matrix and will thus be invertible[5]. This implies that the ELM can estimate training samples with zero error in such a case. Typically, though, a $\hat{N} \ll N$ and consequently, **H** is a non-square matrix [46]. In such a case, it may be that no β exists that exactly satisfies equation 3.3.1 and instead, the minimum norm least-squares solution $\hat{\beta}$ is found using pseudo-inversion as below:

$$\beta = \mathbf{H}^{\dagger}\mathbf{T} \tag{3.19}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalised inverse of **H** [46].

3.3.3 Singular Value Decomposition and Regularisation

As discussed in previous sections, regularisation is a particularly important concept used to preventing over fitting and ensure robust generalisation of performance across out-of-sample data. Singular value decomposition presents an elegant means of computing the Moore-Penrose generalised (pseudo-) inverse as shown in equation 3.3.2. According to [46], it can generally be used to perform this calculation in all cases - even when methods such as orthogonal projection and the iterative method fail. Singular value decomposition applied to matrix $\mathbf{H}_{[m\times n]}$ yields

$$SVD(\mathbf{H}) = \mathbf{USV^T} \tag{3.20}$$

where **U** and $\mathbf{V}^T$ are real valued, orthogonal matrices such that

$$\mathbf{UU}^T = \mathbf{I} = \mathbf{VV^T} \tag{3.21}$$

where **I** is the identity matrix. Consequently, the generalised inverse of a matrix **H** can be calculated easily as

$$\mathbf{H}^{\dagger} = (\mathbf{USV^T})^{-1} = \mathbf{U}^T\mathbf{S^{-1}V} \tag{3.22}$$

Since **S** is a diagonal matrix, its inverse $\mathbf{S}^{-1}$ is also diagonal but on its diagonal elements, merely contains reciprocals of those of S, i.e. diag$(\frac{1}{s_{11}}, \frac{1}{s_{22}}, ...)$ where $s_{11}, s_{22}...$ are the diagonal elements of **S**. In order to prevent numerical ill-conditioning in computation of equation 3.22, a small value λ is typically added to the diagonal elements of **S**. However, this is almost always done as a means of efficient regularisation regardless of potential ill-conditioning since with a large number of hidden neurons required for adequate representational power, such models will often over fit the training data [40]. Regularisation can be achieved efficiently by adjusting the singular value matrix **S** as follows

$$\mathbf{S} \longrightarrow \mathbf{S} + \lambda\mathbf{I} \tag{3.23}$$

[5] provided **H** is non-singular

where λ is the termed the ridge factor and can be optimised iteratively or tuned through LOO (leave-one-out) testing [40]. Singular value decomposition is thus a powerful tool as it allows for fast evaluation of prediction accuracy over a wide range of λ values without having to perform any additional matrix inversions since only $(\mathbf{S} + \lambda\mathbf{I})$ needs to be updated and the pseudo-inverse reevaluated through simple matrix multiplication as in equation 3.22. This allows for exceptionally rapid optimisation of λ [40].

Chapter 4

Design Methodology and Experimental Procedure

4.1 Design Methodology

This project consisted of several major stages that, while all interdependent, required different broad skills. The design methodology in this project was broken down into the following main subsystems:

- Mechanical design of the force plate
- Electronics and hardware pertaining to the data acquisition system
- Embedded software required to interface with the electronic hardware
- Design of the software used to unpack and analyse the data obtained

4.1.1 Conceptual Design and System Specifications

In each broad section of development as mentioned above, a conceptual or overview design was conducted using knowledge developed from the literature review. This process was initiated to properly establish the system specifications and to compare various components or mechanisms that would be best suited to achieve these specifications. In the case of software design - both at the embedded and computer-based post-processing levels - different strategies were first considered and then implemented accordingly.

4.1.2 Detailed Design

Once components and design strategies were established, a detailed design of each subsystem was performed. For the electronic hardware subsystem, this entailed the development of prototypes with the selected components and thereafter, detailed PCB designs. Software design began with a broad overview of the functionality required and was then refined to the system diagram presented in Figure 7.8 which clarified the intricacies of the system. Finally, a fully incorporated and tested software suite was created. These processes were specific to the subsystem in development and are explored in subsequent design chapters.

4.2 Experimental Procedure

This section section seeks to provide insight into the techniques and procedures used to test the data acquisition system and and various aspects of the incorporated force plate system.

4.2.1 Simulations

The system designed in this project was, to a large extent, a novel one and no existing data pertaining to the unique hardware configuration was available prior to development of the operational data acquisition system. As such, few simulations could be performed until data was gathered from the system. However, a FEM stress analysis was conducted on the mechanical structure of the plate using Dassault Systmes SolidWorks in order to gain greater insight into the platform's simulated response to expected loading conditions.

4.2.2 Data Collection and Training

Once the data acquisition system was complete and tested to be operating reliably, a set of systematic experiments were devised in order to thoroughly test the system. Broadly speaking, experiments performed on the plate were conducted order of perceived complexity. In each test, a working 3-axis capacitive force sensor (the OptoForce OMD-45-FH-2000N), hereinafter referred to as the OF sensor, was used to generate truth data - a reference of the actual GRF targets to be estimated by the plate. More detailed specifications of this sensor are given in Appendix D.2. A diagram of the sensor extracted from [48] is shown in Figure 4.1 below.

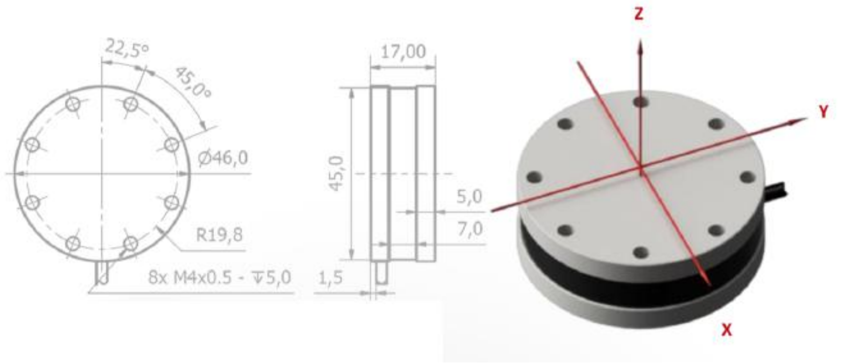


Figure 4.1: Diagram of the OptoForce OMD-45-FH-2000N 3-axis force sensor from [48]. Dimensions of the device are indicated in mechanical drawings of the top and side views thereof.

Single axis testing

Initially, only a single axis vertical force was applied to a single point on the plate and data gathered accordingly. First, only a static vertical load was first applied by placing a weight of known mass on the plate. Thereafter, more dynamic and random vertical forces were applied through the used of a drill press and a rubber foot as shown in Figure 4.2 below. The press was used in order to provide a truly isolated vertical force devoid of any lateral components that would potentially obscure the z-axis data captured by the system.

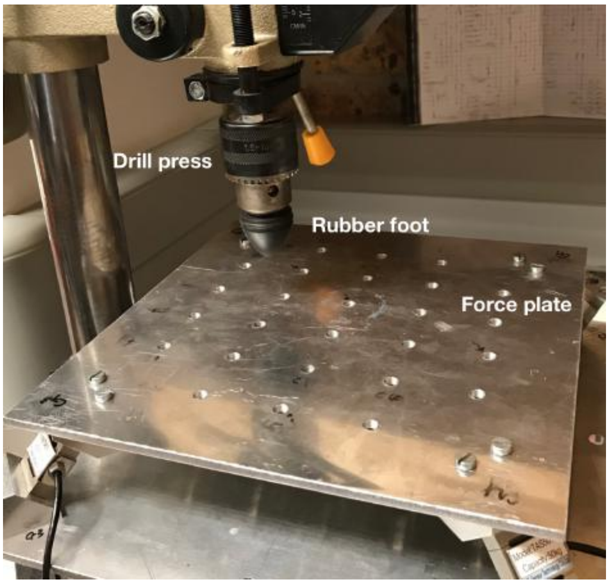


Figure 4.2: Image of the experimental setup used for gathering data during single axis (vertical) force tests. The switches were not yet installed at the time of taking this photograph.

Once performance was tested and accepted for experiments performed on a single point on the plate, data was gathered using the setup presented in Figure 4.2 over a grid of 26 different positions on the plate in order to gather data used to test the robustness of the system in its ability to estimate forces applied regardless of the COP position (position of the applied force).

Two axis testing

The next set of tests performed subjected the plate to a two axis force applied by the drill press. Specifically, a force with purely vertical and mediolateral components (F_z and F_x) was applied during these tests. This was achieved by rotating the mounting platform of the drill press by $\theta = 30°$relative to its zero position[1], as depicted in Figure 4.3

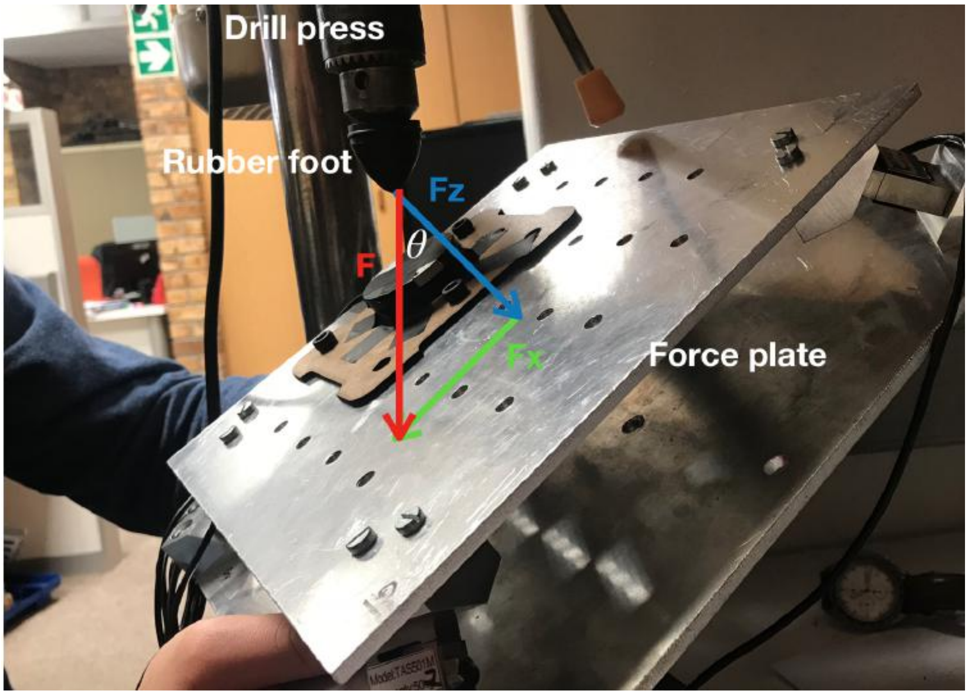


Figure 4.3: Image of the experimental setup used for gathering data during two axis GRF experiments. Vertical (F_z) and mediolateral (F_x) GRF component vectors are shown together with the resultant GRF vector F.

Since the OF sensor was not mechanically fixed to the top of the force plate, it was susceptible to sliding (translation) across the plate when lateral forces were introduced. In order to prevent this undesirable lateral translation, the OF mounting brace depicted in Figure 4.4 was designed and laser-cut. This brace was used in all multi-axial experiments for reasons mentioned above. Note that the switches were temporarily removed to allow the mounting brace to be secured to the plate via bolts in the switch holes.

[1] when the mounting platform is perfectly perpendicular to the vertical telescopic direction of the drill chuck

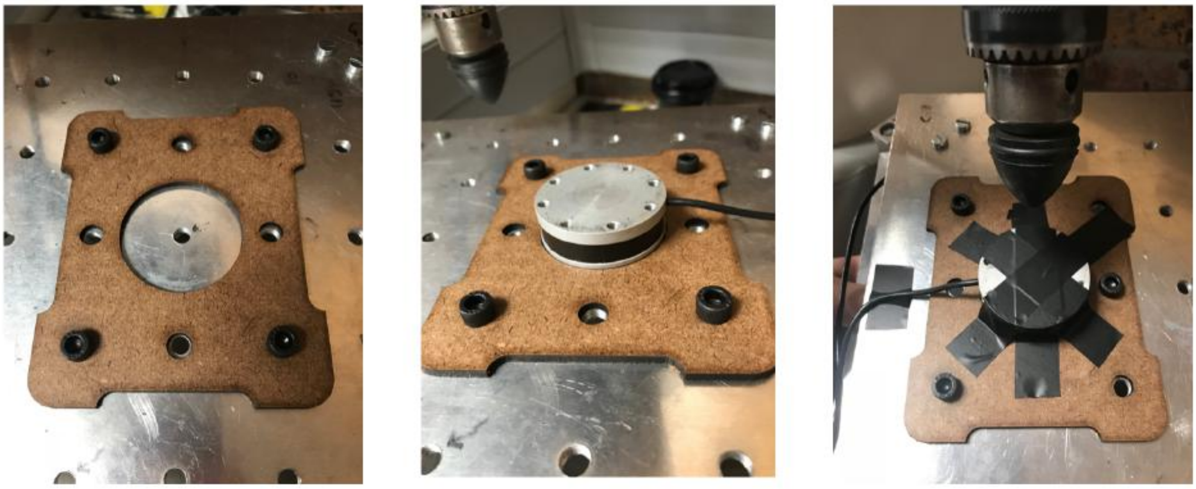


Figure 4.4: Images showing the OF sensor mounting brace used to prevent unwanted lateral *translation* during application of multi-axial forces. The right-most image shows the application of insulation tape used to prevent unwanted *rotation* of the OF sensor.

Three axis testing

Finally, full three axis tests were performed. As with the procedures followed in previous sets of tests, data was first collected and analysed on a single point (the point closest to the centre of the plate) and subsequently, across the entire grid of points on the plate. Initially, training data was collected by applying gradual three-axis forces using a test rod (pictured in Figure 4.5 below) comprised of a metal rod and removable rubber foot (same as used in the drill press tests).

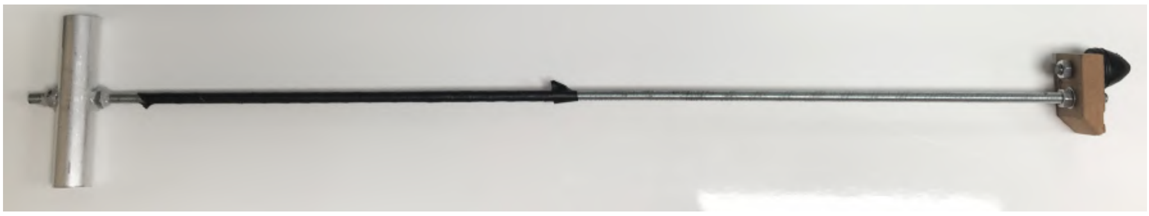


Figure 4.5: Image showing the test rod used to generate gradual and more consistent 3-axis force data

Figure 4.6 below depicts a profile of the force components applied to the plate using the testing rod during experiments to gather training data. Notice the very gradually increasing and decreasing force patterns - time axis is in seconds and most minimum to maximum transitions occur over a matter of seconds where several thousand samples could be taken. The significance of this is explored in Section 10.2.4.

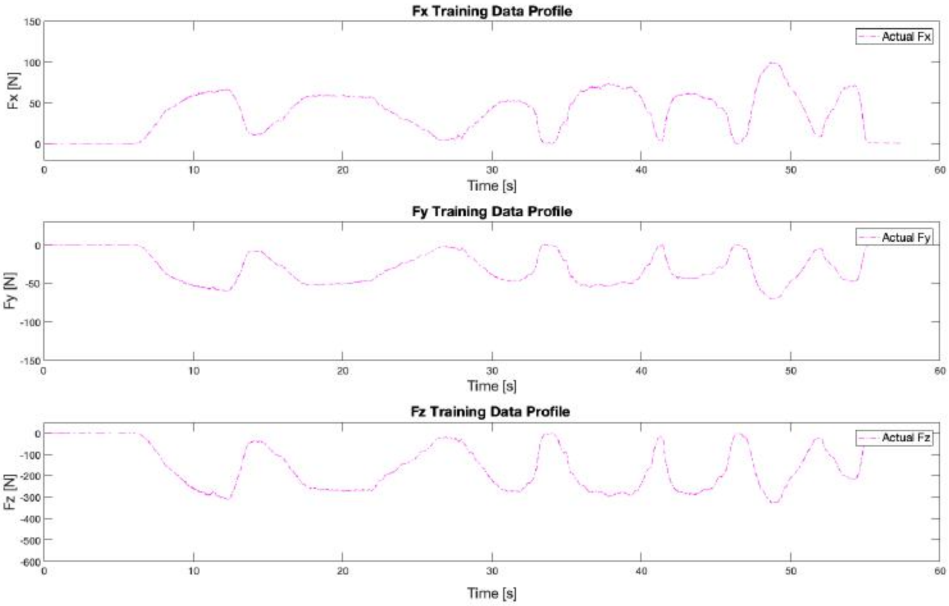


Figure 4.6: Graph showing a typical training data profile used in 3-axis experiments

Subjects were then requested to walk and run over the plate in a natural fashion to gather more impulsive data. Subjects were requested to walk primarily along the y-axis but were instructed to allow for random lateral (x) motion during push off to introduce all three component forces into the data. Note that the OF sensor was still required to be secured in place by the mounting brace in order to prevent translation (slipping) of this sensor during tests. As such, the switches were left disconnected during the collection of training data in order to enable secure mounting of the brace. The operation of the switches was tested and verified independently of this test.

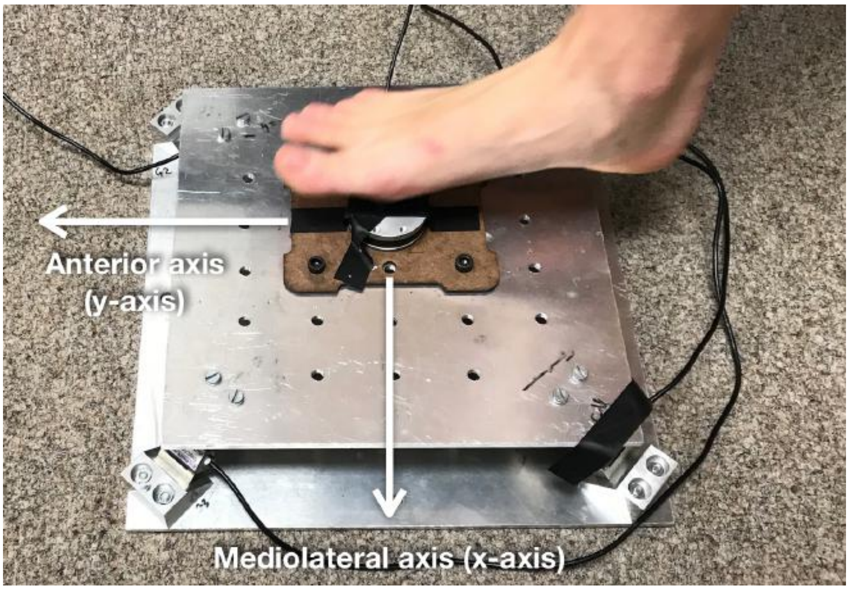


Figure 4.7: Image captured during a walking test showing a subject about to step on the plate

Chapter 5

Mechanical Design

The mechanical design of the force plate used in this project is largely a novel configuration and was proposed to the author by Dr Amir Patel and Mr Callen Fisher who, amongst others, applied for a patent on the mechanical design shortly prior to the time of writing of this report. As alluded to in Section 1.3, a detailed analysis of the mechanical structure proposed and/or an investigation into alternate mechanical designs do not form part of the scope of this project. Rather, a brief analysis of the proposed design is explored and the rationale behind it clarified.

5.1 Proposed Conceptual Design

The motivation explained above led to a design that requires only four single-axis load cells as the primary force sensors. The novel configuration proposed by Patel, Fisher *et. al.* is clarified by the 3D render presented in Figure 5.1 below. The design consists of two square plates - a base bottom plate and a smaller upper plate with centres vertically aligned. The plates are coupled through four load cells mounted at predefined angles; allowing the load cells to respond to components of forces applied to the top plate in all three axes. The geometric arrangement of the four load cells is depicted in figures 5.1[1] and 5.6 below. A grid of 26 threaded holes was tapped into the top plate in order to house a matrix of switches used to determine the position of the COP during GRF interactions (the rationale for this is explored in subsequent sections).

Both the top and bottom plates, as well as the load cell mounts, were cut and/or machined from 6061-T4 heat-treated aluminium alloy with a yield stress of 240 MPa and a Young's modulus of 68.9 GPa [47]. This material was selected due to the fact that it is relatively inexpensive, light-weight and robust.

[1] Credit is given to Alexander Blom of the UCT Mechatronics Lab for the SolidWorks models of the load cells and angled mounts

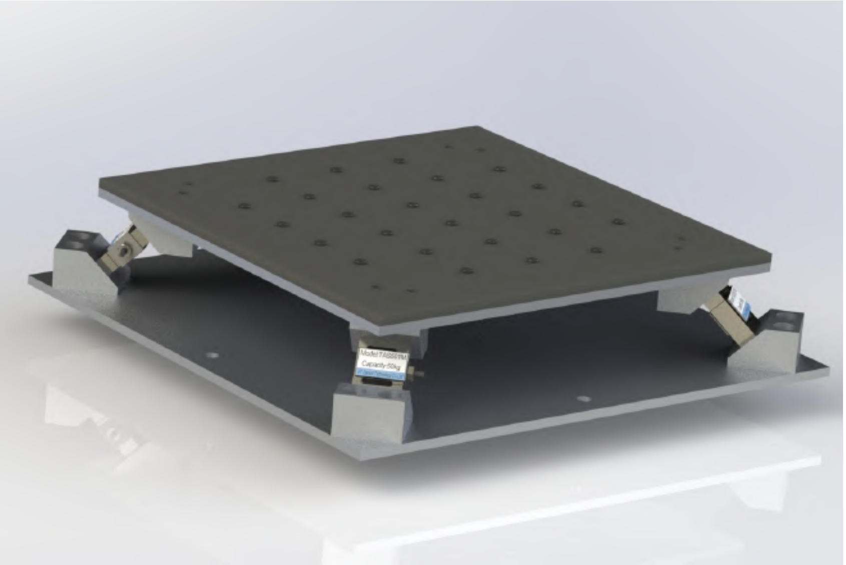


Figure 5.1: Render of the mechanical design of the force plate proposed by Patel, Fisher *et. al.* with modifications from the author.

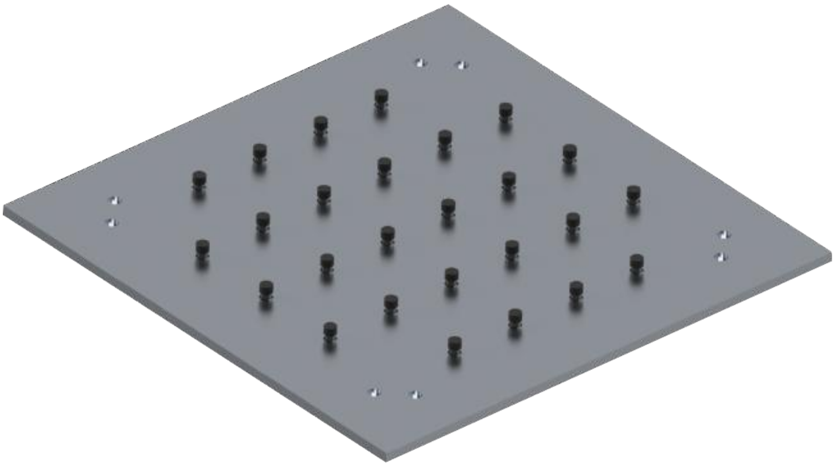


Figure 5.2: Render of the top aluminium plate with the 26 push switches mounted as per the co-ordinate scheme in Figure A.1

5.1.1 The Mechanical Challenge of the Design

The challenge presented by the aforementioned design is that, from a solid mechanics perspective, it poses a difficult system to solve analytically due to the fact that each load cell is rigidly connected (bolted) to the top and bottom plates through inclined mounts. Since the load cells are only designed to measure in-axis compression and tension forces, it is difficult to discern the effect of torque-inducing shear forces[2] on the differential voltage produced by each gauge. As such, it is extremely hard to derive component forces measured by the load cells directly and instead, a data-driven numerical approach was adopted to tackle the problem.

[2] forces perpendicular to the primary longitudinal axis of the load cell in which compression and tension are measured

5.2 FEM Simulation

A series of FEM (finite element method) stress analysis tests were performed using SolidWorks in order to verify the the mechanical design. As it was identified that the top plate was likely to be the point of first mechanical failure, it was analysed in detail. In the simulation, all corners of the top plate were fixed (as if mounted by bolts in their actual configuration) and a force was applied at the centre of the plate. A small hole at the centre of the plate can be seen in figures 5.3 and 5.4 - this served as a work around to simulate a point force at the centre of the plate since the software used only allowed for *distributed* forces to be created on a face (plane). This small circle offered a new, sufficiently smaller plane to more closely simulate a point force.

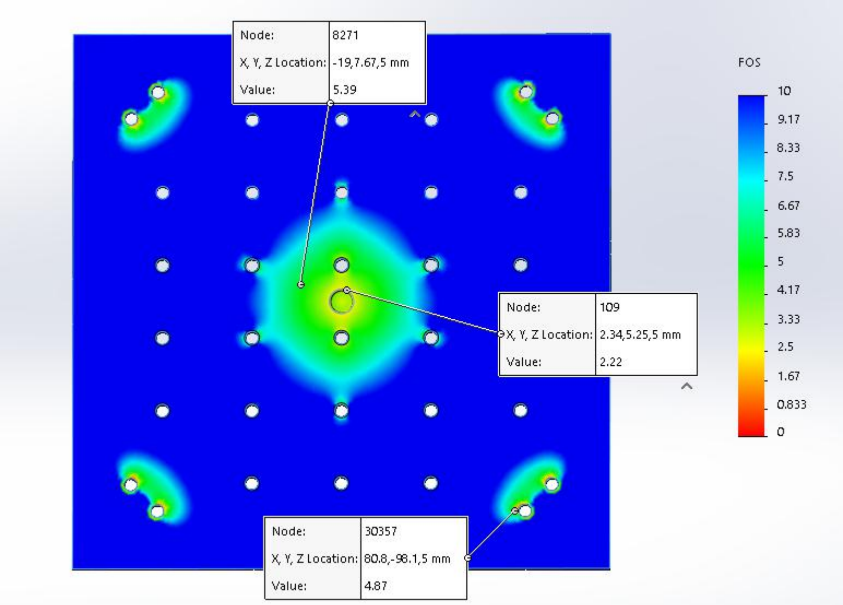


Figure 5.3: Diagram showing the FOS (factor-of-safety) results of a FEM analysis with a 1000N normal load and a 150N mediolateral load applied at the centre of the plate

As seen in the factor-of-safety (FOS) plot in Figure 5.3 above, 2.22 times the applied load of 1000N in the normal axis and 150N in the mediolateral axis can be tolerated by the top aluminium plate before yield stress is reached and deformation occurs. However, this is likely a conservative estimate since the maximum simulated stress was located on the edge of the temporary hole in the centre of the plate which does not actually exist in the design. Observing the rest of the simulation result, an FOS of approximately 4.5 is more likely closer to the actual rating and thus, it is safe to conclude that material failure is not of major concern for the loading conditions expected for this system (both in human and animal testing applications).

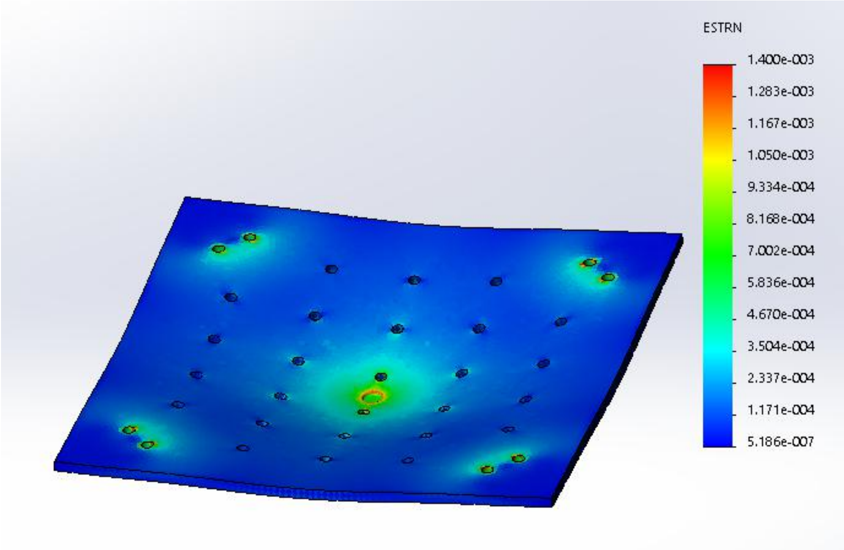


Figure 5.4: Diagram showing the material strain results of a FEM analysis with a 1000N normal load and a 150N mediolateral load applied at the centre of the plate

5.3 Implementation of the Proposed Design

A picture of the constructed platform, as per the design shown in Figure 5.1, is shown in Figure 5.5 below. As shown in the render in Figure 5.2, secure mounting of the switches in the top aluminium plate resulted in them protruding considerably above its top surface. A 3mm thick neoprene rubber sheet was secured to the top aluminium plate so that upon foot contact with the plate, the switches were only allowed just enough travel to successfully trigger, but not enough to sustain any mechanical damage.

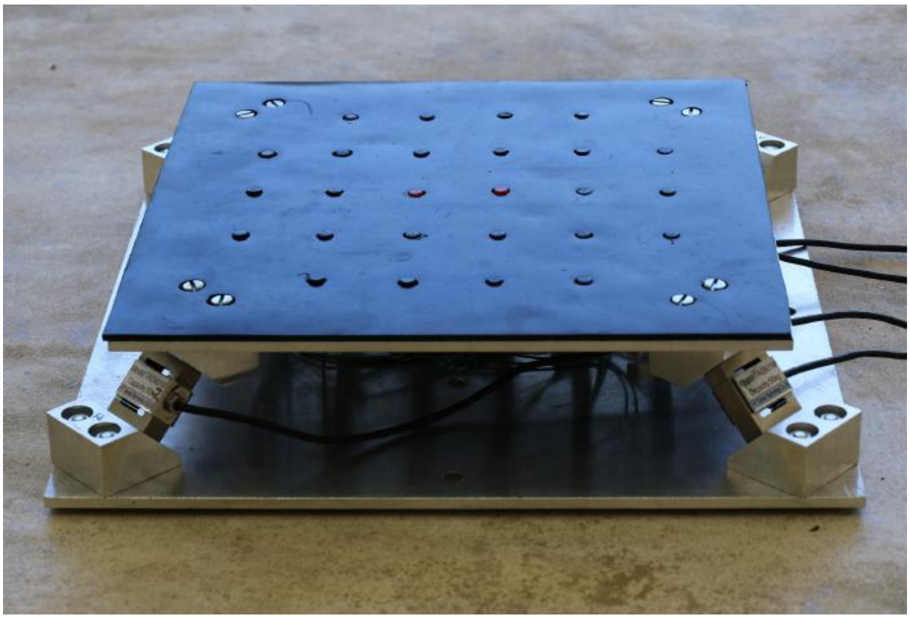


Figure 5.5: Image of the constructed force plate. Notice the carefully adjusted protrusion of the switches.

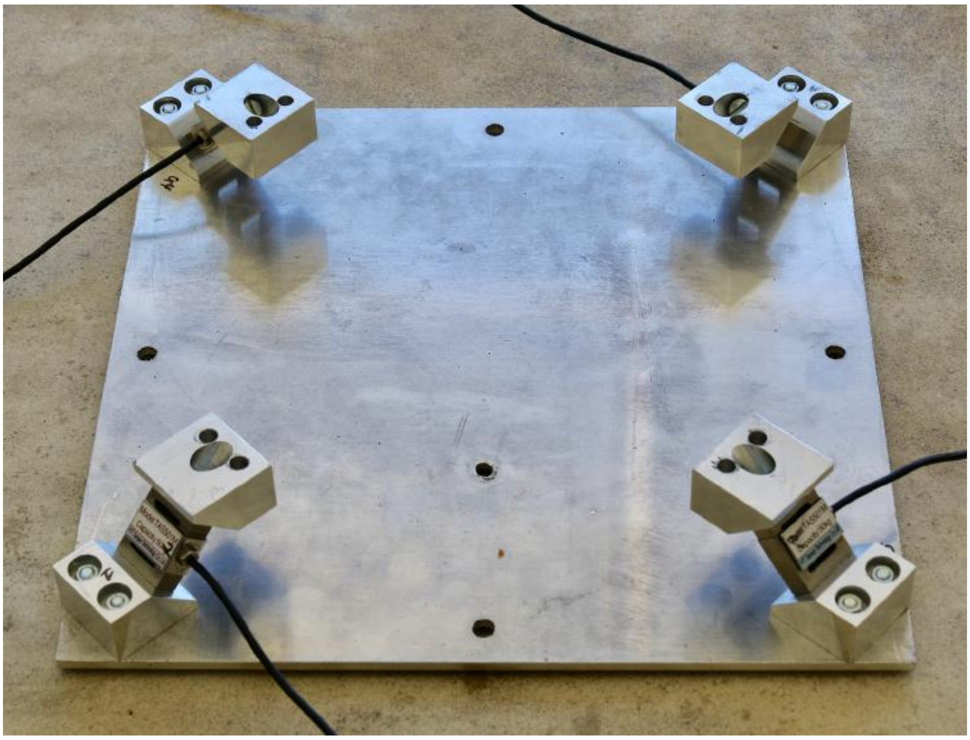
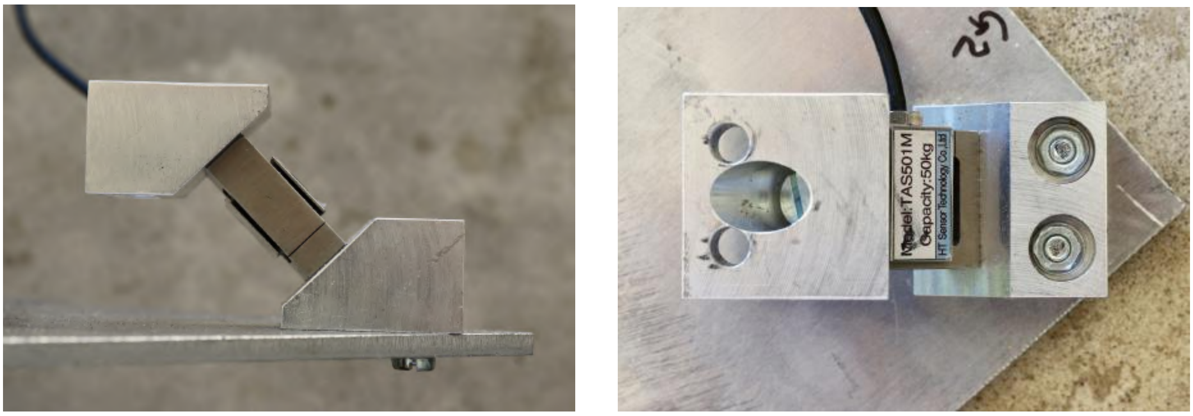


Figure 5.6: Image of the constructed system with top plate removed to show the load cell geometry.



(a) Side view

(b) Top view

Figure 5.7: Images of one of the four identical load cells mounted at 45°relative to the horizontal plates.

Chapter 6

Electronic Hardware Design

One of the most critical subsystems in this project is the data acquisition system. When adopting a data-driven machine learning approach as in this project, there is a particularly strong reliance on the quality of data acquired from all sensing elements. The concept of GIGO (garbage in, garbage out) is particularly relevant here - one can only expect useless results from a system that is provided useless input data. As such, this section explores the electronic hardware aspects of the data acquisition system designed and used in this project. Development begins with a conceptual overview of the system and its specifications and then progresses to a comparison and justified selection of hardware components. Finally, the selected components are explored more thoroughly and the design and development of PCBs is explored.

6.1 Conceptual Design and Component Selection

The data acquisition system to be used in this project had the following fundamental requirements:

- Capable of sampling load cells at up to 1kHz
- High fidelity and ultra low-noise
- Robust and tolerant to non-ideal power supplies

The first requirement of 1kHz sampling ability was guided by findings in the literature (Section 2.3.3) and was identified as a particularly important requirement of the system. It was realised that in order to have the ability to log data at 1kHz for several seconds, several hundred kB of memory would be required. Consequently, storing data directly in the RAM of a micro-controller was no longer an option and the need for an SD card logging interface was identified. Figure 6.1 below shows a broad overview of the envisaged data acquisition system:

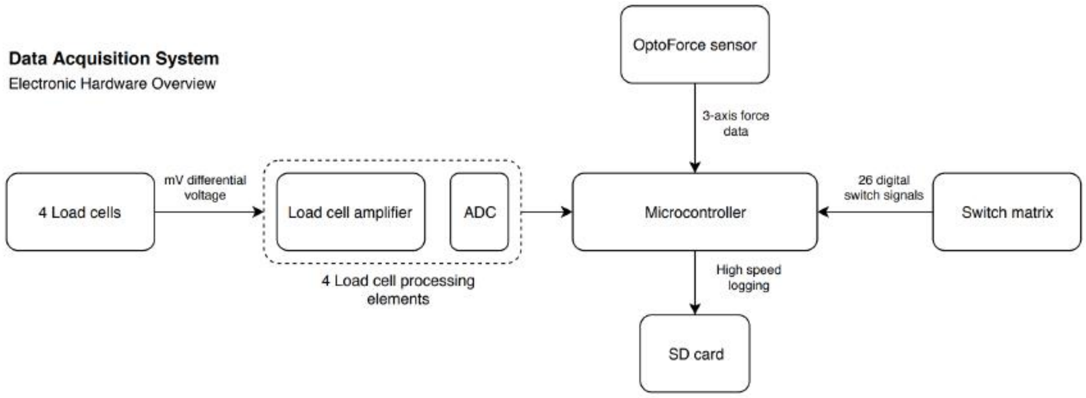


Figure 6.1: Diagram of the conceptual hardware overview of the data acquisition system

As shown above, the data inputs received by the micro-controller include readings from the four load cells, data packets from the OptoForce sensor and digital signals from the 26 switches mounted in the top plate. The micro-controller was responsible for receiving data from the aforementioned sources, processing and packetising the received data and finally, logging packets to the SD card. These constituent hardware components are explored separately below.

6.1.1 Load Cell Signal Processing

One of the most important components in the data acquisition system is the component responsible for amplifying and sampling extremely small differential voltages from the load cells. Each of the load cells used in this project were TAS501M miniature S-type load cells made by HT Sensor Co. with a maximum capacity of 50kg and rated output of 1.5 ± 0.5 mV/V. Consequently, for an excitation voltage of 5V, these load cells would produce a theoretical[1] maximum differential voltage of only $V_{d(max)} = 7.5$mV at the terminals of the full Wheatstone bridge embedded within them.

Considering that such a small differential voltage range was available, noise introduced by the surrounding environment, amplifiers and interfacing circuitry and importantly, transmission to the micro-controller, became a very important consideration. It was recognised that external sampling and digitising of the signals from the load cells posed a superior solution to reading analogue load cell voltages directly into the micro-controller. This was decided for the following reasons:

- Compared to analogue signals, digital signals are extremely noise-resistant as they encode information in binary voltage levels
- Specialised ICs exist that are designed for ultra low noise amplification and sampling of small scale differential voltages. Such ICs are explored below.

[1] this is an ideal maximum that neglects standard error and potentially, the inability to reach the full scale excitation voltage

- Four independent high-precision ADCs are required to truly be able to sample all four load cells at the same instant. Most cheap micro-controllers do not have four internal ADCs.
- The data acquisition system becomes decoupled from the specific micro-controller used if external amplification and sampling is used and increases modularity and cross-compatibility with different micro-controllers is possible

Component comparison

The HX711, made by AVIA Semiconductor, is an IC designed for weigh scale and industrial process control applications and offers an integrated low-noise differential amplifier with programmable gain, 24-bit sigma-delta ADC and a proprietary digital interface. It is extremely popular in hobbyist applications and its use is documented extensively across the internet.

Another largely similar IC designed for the same purpose is the AD7730 made by Analog Devices. It offers all the same internal functionality as the HX711 above with the addition of two on-board digital filters, a more universal SPI output interface and a 6-bit DAC to offset (tare) biased input voltages (among other minor additions). However, while supported by extensive documentation by Analog Devices, the AD7730 is largely undocumented across the internet and in hobbyist applications.

In an attempt to realise the most suitable option for this project, the comparison depicted in Table 6.1 below was compiled with reference to the component datasheets in [49] and [50]:

Table 6.1: Table comparing the specifications of the AD7730 and HX711

	AD7730	**HX711**
ADC resolution	24-bit	24-bit
Digital interface	SPI	Proprietary pulse-read mechanism
PGA options	Continuous (0-255)	Discrete (32, 64 or 128)
Offset DAC	Yes	No
Calibration	Full offset and gain calibration options	None
Digital filtering	Programmable digital filters	Fixed dual 50/60Hz rejection filters
Maximum ODR	1.2kHz	80Hz
Output noise @ 80Hz ODR	67nV	90nV
Input common mode rejection	100dB	100dB
Cost	$12.91	$9.95

The most glaring difference exposed by the comparison in Table 6.1 above is the maximum

ODR (output data rate) supported by the two devices. The HX711 only allows for a maximum ODR of 80Hz and this alone renders it unsuitable for this project. As a result, the AD7730 was selected for use in this project and is explored further in Section 6.2.1 below.

6.1.2 Data logging

Two main options were identified to interface with the SD card for high speed logging of data: the Sparkfun OpenLog SD card logger which operates via UART and a generic SPI SD card breakout board. These devices are pictured in Figure 6.2 below.

(a) Sparkfun OpenLog micro SD card breakout [59] (b) Adafruit micro SD card SPI breakout [58]

Figure 6.2: Images of the micro SD card logging devices considered

While both devices are capable of sufficiently fast data transfer rates, the main distinguishing feature between them is the complexity required to physically write serial data to the SD card. In order to write data to the SD card using the Adafruit SPI device, a FAT32 file system implementation is required. Although FATFS, an open source project that provides high level implementations of the FAT file system , is available, it requires significantly more complexity in the embedded software. Comparatively, the OpenLog device has a separate ATmega328 micro-controller pre-flashed with FAT32 firmware and as a result, data can simply be logged to the SD card through UART communication. For this reason, the OpenLog was chosen for the purpose of data logging.

6.1.3 Micro-controller

The micro-controller used in this project was the STM32F407VGT6 Discovery featuring a 32-bit ARM Cortex M4 core with FPU, 1Mb flash memory and 192Kb RAM. Relevant to this application, this micro-controller also has 3 SPI and 3 USART peripherals, as well as an on board 32-bit hardware CRC unit (the significance of which is discussed in Chapter 7).

It was recognised that a high performance micro-controller capable of coordinating receiving, computing and packetising and sending data packets in under 1ms (in order to achieve 1kHz sampling) was required. As a candidate that more than meets the performance requirements, STM32F4 Discovery was chosen for convenience (the fact that it is a development platform and not just a stand alone MCU) but also due to the fact that it is one of the more widely documented micro-controllers with a vast array of resources available on the internet.

6.2 Detailed Design

Once the major components for the functions mentioned above were identified, a more detailed design of the electronic hardware was conducted. Figure 6.3 below shows a refined version of the overview diagram presented in Figure 6.3 in greater detail. The subsystems shown in Figure 6.3 - such as the AD7730 and shift register interface for the switch matrix - are explored separately below.

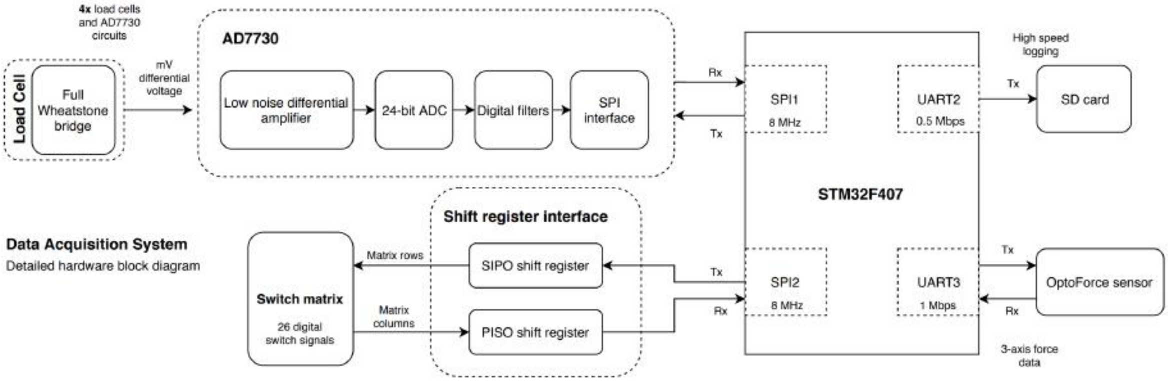


Figure 6.3: Diagram showing a more detailed hardware block diagram of the electronics comprising the data acquisition system. Note that while not depicted explicitly, the system utilised four load cells and four corresponding AD7730 chips which operated identically.

6.2.1 AD7730 for Load Cell Interfacing

As mentioned in Section 6.1.1 above, the AD7730 is a highly specialised IC made by Analog Devices and is designed for sensitive weigh scale and pressure measurement applications offering an offset drift of less than 5nV/°C and a gain drift of less than 2ppm/°C. A functional block diagram of the device is presented in Figure 6.4 below.

When using this device in DC-excited bridge applications, such as with Wheatstone bridges as used in this system, a DC excitation voltage is applied to the bridge and is used as the analogue reference voltage seen by the on-board ADC. The mV differential voltage produced by the Wheatstone bridge in each load cell is buffered to prevent the effects of loading on the resistive bridge and then amplified by an ultra-low noise 8-bit PGA which can be programmed to any continuous gain between 1-256. The device leverages a 24-bit sigma-delta ADC which digitises the PGA output with a peak-to-peak resolution of 1 in 230 000 counts [49], although this is the quoted performance straight from the output of the device and noise in the transmission and surrounding environment will lead to a decreased noise-free resolution. The device also features an internal calibration micro-controller and is capable of performing self-calibration (without

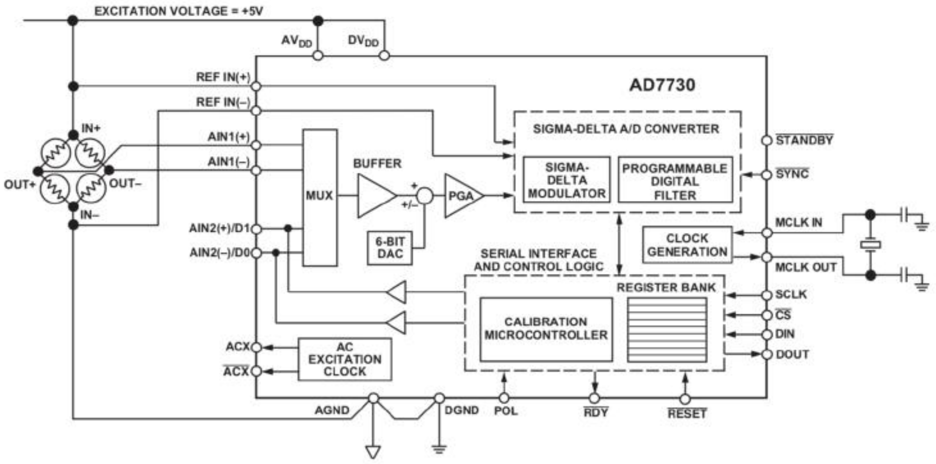


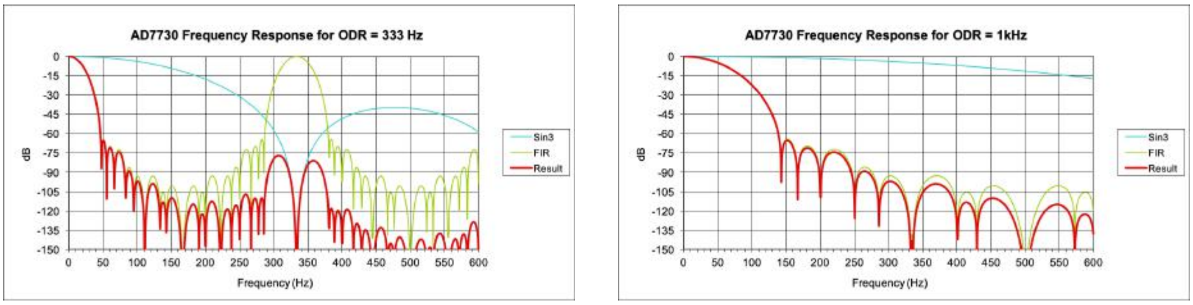
Figure 6.4: Diagram showing a functional block diagram of the AD7730 from [49] connected in a typical DC-excited bridge application (as used in this project).

the need for an external differential input) and system calibration (using the zero- and full scale external differential inputs).

On-chip digital filtering

A particularly notable feature of the AD7730 is its on-board digital filtering capability. The digitised output signal from the ADC is fed into a digital sinc^3 low-pass filter with programmable ODR and bandwidth (as explored in Section 7.4 below). The sinc filter receives its name from the fact that it is a $\text{sinc}(\omega) = \frac{\sin(\omega)}{\omega}$ function in frequency and a rectangular pulse (ideal low-pass filer) in the time domain. In the AD7730, the ADC samples the differential input from the PGA at a considerably higher frequency than the prescribed/desired final ODR (and the bandwidth of the input signal) and then, the sinc^3 filter performs averaging on the over sampled ADC signal by down-sampling to the rate specified by the prescribed ODR. As explained by [51], sinc filters are used almost universally for this downsampling of sigma-delta ADCs due to their simple implementation and near-optimal performance. The primary function of this is to remove quantisation noise introduced by the ADC modulator that can be significant [49].

The second filtering stage is performed by a 22-tap low-pas FIR filter which processes the downsampled output of the sinc^3 filter. The combined frequency response is merely the product of the filter responses of both filters.



(a) Responses for ODR = 333Hz

(b) Responses for ODR = 1kHz

Figure 6.5: Diagram of filter frequency responses (magnitude) for two different ODRs. The sinc^3, FIR and combined filter responses are shown.

As seen in Figure 6.5 above, the combined two-stage filter response is significantly influenced by the programmed ODR of the device. It is evident from Figure 6.5(b) that high frequency sampling at 1kHz is limited by considerably lower attenuation of line frequencies; the combined frequency responses provides only -5dB and -7dB attenuation at 50 ± 1Hz and 60 ± 1Hz respectively. However, as reflected by Figure 6.5(a), a lower ODR of 333Hz yields a combined attenuation of -66dB and -71dB attenuation at 50 ± 1Hz and 60 ± 1Hz respectively. It should be noted that the FIR filter in the AD7730 can also be bypassed completely in order to capture additional high frequency components (although, significantly increased susceptibility to high frequency noise would also need to be considered in such a case).

Circuit design

Once a greater understanding of the AD7730 was developed, a test circuit was devised to begin testing of the micro-controller SPI interface library described in Section 7.4. Figure 6.6 below shows a schematic of the AD7730 interface circuit used to communicate with the micro-controller. Note that in the implementation of the prototype board, AGND and DGND were connected directly at several points (unlike in the PCB design explored below).

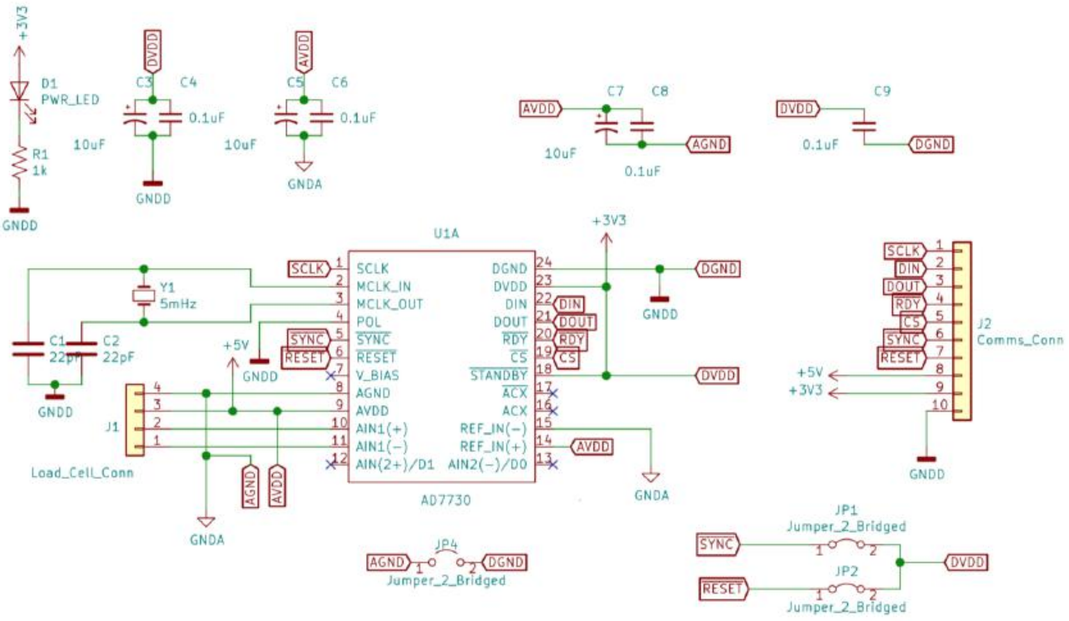


Figure 6.6: Schematic of the AD7730 interface circuit

The key objectives of this circuit were to provide a means of exposing or 'breaking out' the pins of the chip to allow for interfacing with the micro-controller. As per the AD7330 data sheet [49], an independent 5MHz crystal oscillator was placed on the circuit for each chip to allow the crystal to be placed as close to the external master clock pins of the AD7730 as possible in an attempt to reduce noise. Apart from the regular SPI pins (SCLK, DIN, DOUT and $\overline{\text{CS}}$), it was decided that the $\overline{\text{SYNC}}$, $\overline{\text{RESET}}$ and $\overline{\text{RDY}}$ pins should be available for a hardware connection to the micro-controller, even though the functionality offered by these pins can largely be implemented using software as discussed in Section 7.4. This was done in case unforeseen difficulties were to arise in the software implementation but also in an attempt to design a more universal interface board that could potentially be used by other applications that might require physical hardware connections to these pins. It was also recognised that $\overline{\text{SYNC}}$ pin of the AD7730 can be used to set its ADC and digital filters to a well-defined initial state before conversions begin. This could prove useful in synchronising many AD7730 chips in independent force plate systems in a single experiment. A prototype of this circuit was first built on veroboard and is shown in Figure 6.7 below. Note: All white text labels besides the power labels in the top right of the figure refer to the molex/IDC plug alongside the label.

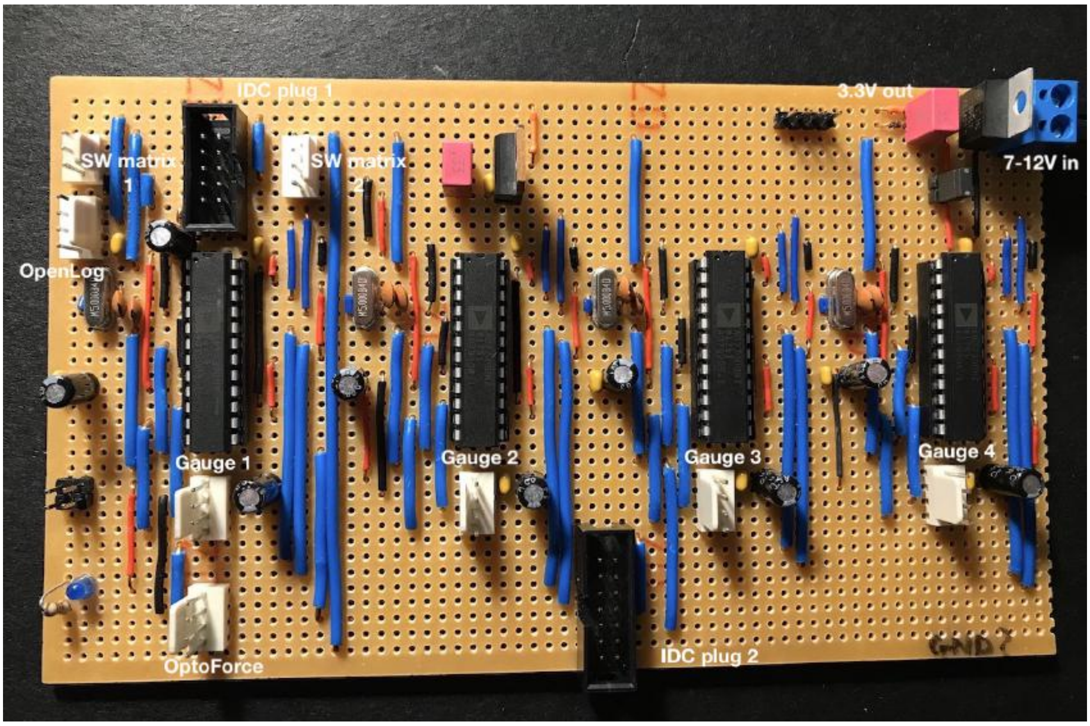


Figure 6.7: Image of the first veroboard prototype constructed to test the four load cells interfaced by four identical versions of the circuit presented in Figure 6.6.

The prototype shown above was also designed, along with housing the four AD7730 chips and plugs for four load cells, to provide connections for the OpenLog micro SD card logger and OptoForce sensor via 4-pin molex plugs for two power lines and UART Tx and Rx lines (for each device). The two plugs labelled SW matrix 1 and SW matrix 2 in Figure 6.7 were added to connect the 7 pins of the switch matrix detailed below. The two IDC plugs labelled IDC plug 1 and IDC plug 2 were used to connect all necessary pins to the micro-controller via ribbon cables.

Initially, the decoupling capacitors shown in the schematic in Figure 6.6 were omitted and for a significant period of time, the circuit behaved extremely erratically and unpredictably. Not only were the analogue readings measured the AD7730 chips noisy and inconsistent, but even digital communication errors began to creep in. Eventually, this mistake was realised and decoupling capacitors were incorporated according to the guidelines specified in the AD7730 datasheet [49]: 0.1μF and 10μF capacitors in parallel connected across the analogue supply lines (AVDD and AGND) and a 0.1μF capacitor across the digtial supply pins (DVDD and DGND). Further decoupling capacitors were also added across the power lines at each molex plug connecting the load cells to the board (labelled ‘Gauge 1’ to ‘Gauge 4’ in Figure 6.7). The introduction of proper decoupling improved the performance of the system dramatically and provided readings with far less noise from the load cells.

6.2.2 Switch Matrix

In order to determine the position of the applied force on the plate (position of the COP), a matrix of 26 momentary switches was created. While the significance of being able to discern the position of the COP in estimating GRFs is explored in subsequent sections, this section explores the hardware designed to provide an interface between the switches and the micro-controller to read and interpret each digital switch signal.

If each switch were to be read independently and directly by the micro-controller, 26 different GPIO pins would be required. This is an extremely inefficient use of the micro-controller's available pins and instead, a system was designed to minimise the number of pins required. A pair of complementary 8-bit shift registers were used - a 74HC595 serial in, parallel out (SIPO) shift register and a 74HC165 parallel in, serial out (PISO) shift register. A matrix of rows and columns was formed by the switches as shown in Figure 6.8 below. This approach allows a grid of up to 8x8 (64) switches to be read using only 5 GPIO pins on the micro-controller.

The SIPO 74HC595 shift register is responsible for writing to the rows of the matrix and the PISO 74HC165 is responsible for reading the state of the matrix columns that default to logic low due to the presence of pull down resistors R1 to R5. A row is set high by sending a byte to the serial input pin of the HC595 with the corresponding bit set and once the whole byte has been shifted and latched into its output data register, the corresponding parallel output pin (row) is set high. The logic level of the columns are latched into the input data register of the HC165 when the SH/$\overline{\text{LD}}$ pin is pulled low momentarily and then returned high. The latched byte is then shifted out after 8 rising edges of SCLK.

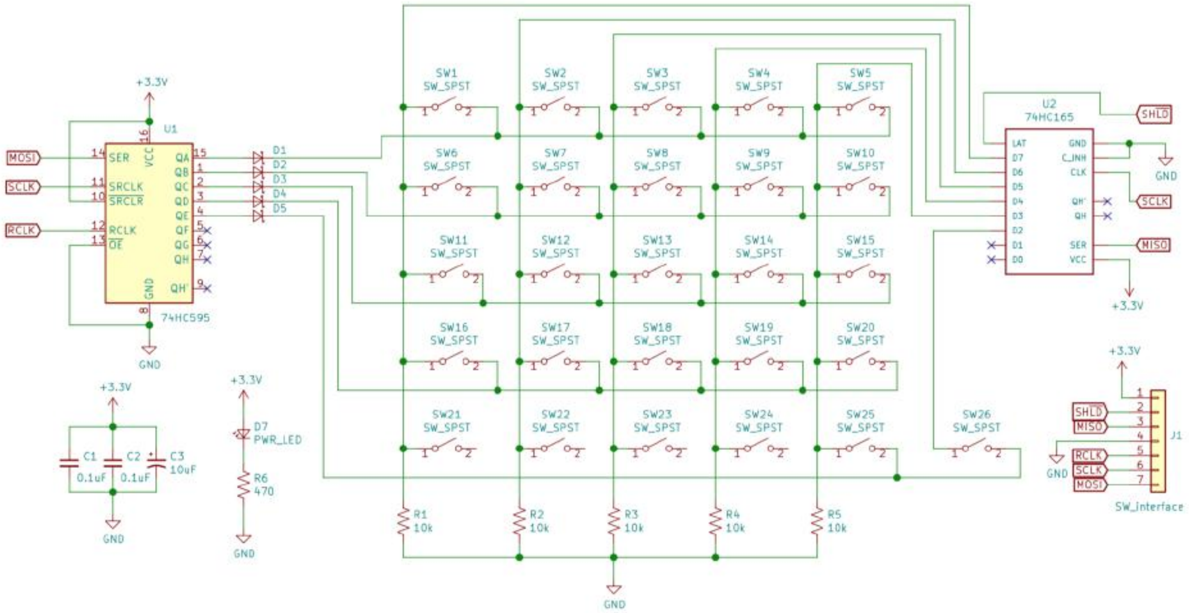


Figure 6.8: Schematic of the switch matrix interfacing circuit showing the 74HC595 and 74HC165 shift registers used to write to the rows and read from the columns of the matrix respectively.

In order to demonstrate the working principle of this circuit, consider row 1, (connected to QA of 74HC595) and only row 1, set high. If any of the switches besides those in row 1 are pushed, there will be no effect as all other rows are at logic low and closing any of these switches would connect a row and column both at logic low. However, if switch 1 in Figure 6.8 (the switch connecting the first column and first row) is pressed, the first column of the matrix will be connected to row 1 at logic high and bit D7 (connected to column 1) of the 74HC165 will be set high. More concisely phrased, this technique allows for the unambiguous deduction that if row i is exclusively set high and column j is read high, switch S_{ij} must be pressed[2].

In the first revision of this circuit, the Schottky diodes were not included and it was noticed that when pressing two buttons on the same column simultaneously, the byte read in by the microcontroller would be incorrect (always zero). In order to demonstrate the need for Schottky diodes D1 to D6, consider the scenario posed before with only row 1 set high and switch 1 pressed, connecting row 1 and column 1. Now consider the consequence of switch 6 being pressed *simultaneously*; column 1, now set high through row 1, would be directly connected to row 2 at logic low. Without Schottky diode D2 present, there would be a short circuit path between rows 1 and 2 (pins Q_A and Q_B of the 74HC595). D2 thus serves to prevent current flowing back into pin Q_B (row 2) and shorting column 1 to ground. Schottky diodes were specifically chosen to minimise the voltage drop from 74HC494 outputs to ensure that the correct columns were still read as logic high where appropriate. While specific rows and columns are referred to here, the principle generalises naturally.

Although only one row can be set high at a time, propagation delays of both shift registers are in the order of ns (nanoseconds) and the SPI interface can be operated in the MHz region. This has the effect of all switches being able to be read simultaneously.

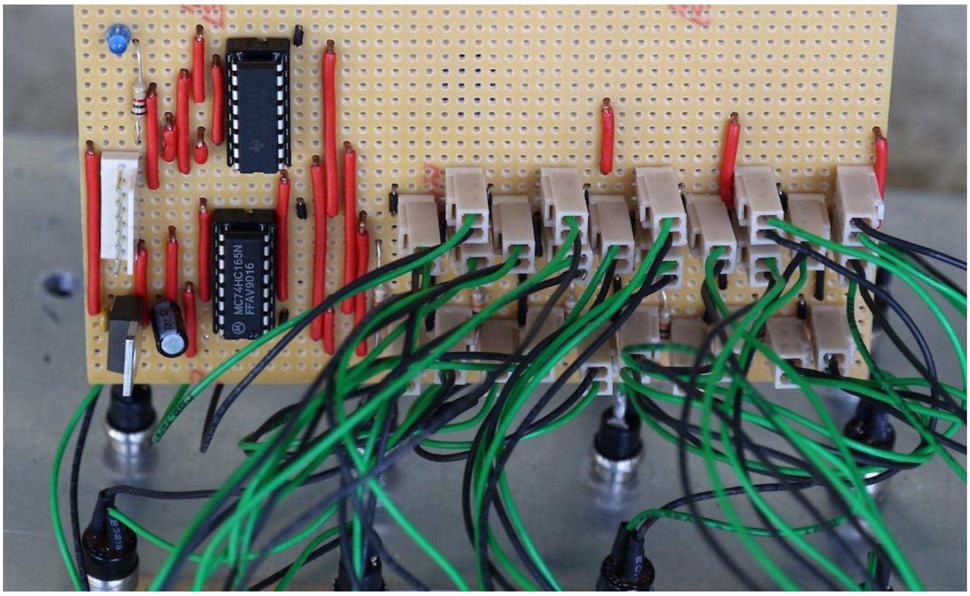


Figure 6.9: Image showing the switch matrix prototype board constructed with reference to the schematic in Figure 6.8.

[2] where switch S_{ij} is located at the junction of the ith row and jth column of the matrix

6.2.3 PCB Design

In designing a printed circuit board for the combined data acquisition system, it was decided that four separate ‘break out’ boards would be developed for the AD7730 ICs, as well as an independent ‘motherboard’ to connect the four AD7730 breakout boards, the OptoForce sensor, OpenLog micro SD logger and the switch matrix to the micro-controller. This approach was taken instead of designing one single board for to house all components for the following reasons:

- This approach offers greater **modularity** and if one component or subsection breaks or is incorrectly designed, it can likely be fixed more easily.
- This approach is more **universal**; the AD7730 breakout boards were designed to not be specific to the application used in this project and can thus be used in other applications.

AD7730 breakout boards

In order to achieve, or come as close as possible to achieving, the impressive performance quoted in the AD7730 datasheet [49], the PCB design guidelines set out in [49] were followed carefully. Extracted from [49] and adapted slightly, these requirements for the PCB housing the AD7730 are as follows:

1. Analogue and digital sections must be separate and confined to certain areas of the board
2. Separate digital and analogue ground planes must be used and must be only joined in one place
3. Digital lines should not be run under the device to prevent noise coupling onto the die
4. The analogue ground plane should be allowed to run under the device to further avoid noise coupling
5. Fast switching digital signals such as the SCLK line should be shielded by digital ground to avoid radiating of noise to other sections of the board. Clock signals should never run near analog inputs
6. Avoid cross-over of digital and analogue signals as far as possible

A 3D render of the AD7730 breakout board designed using KiCAD is presented in figures 6.10 and 6.11 below. The PCB was designed with reference to the schematic depicted in Figure 6.6.

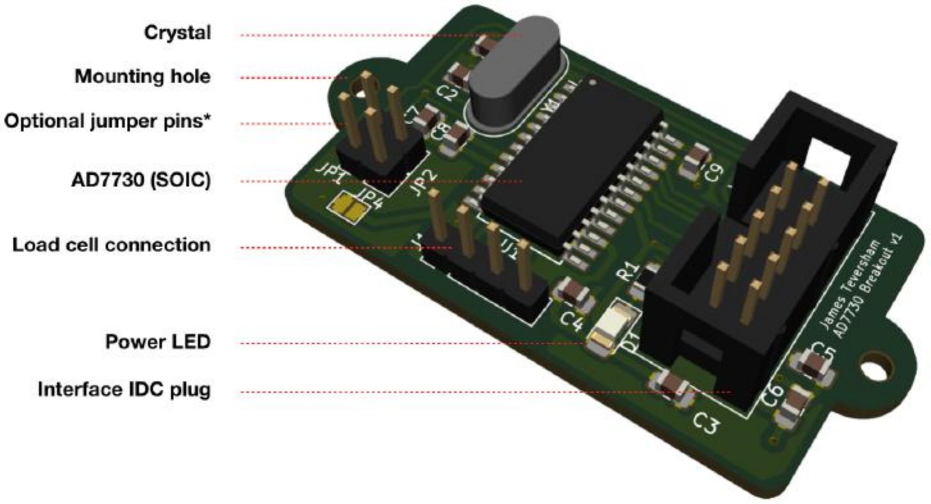
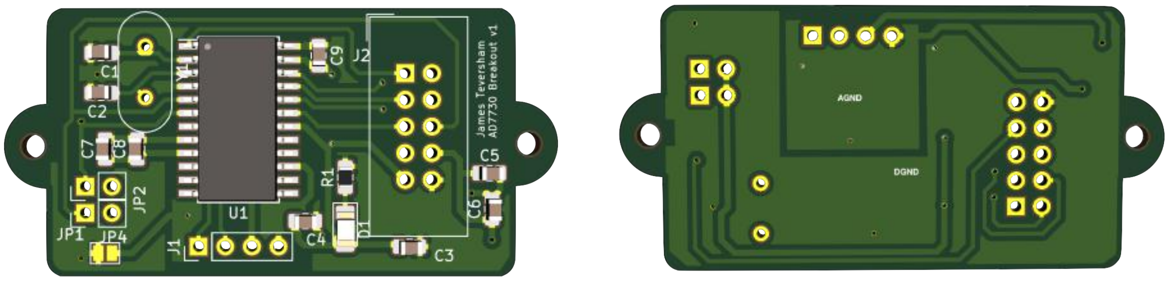


Figure 6.10: Annotated 3D render of the designed AD7730 breakout board with surface mount and through-hole components displayed

Note that in the annotated diagram in Figure 6.10 above, the pins labelled 'optional jumper pins*' can be connected by jumpers in order to pull active-low $\overline{\text{RESET}}$ and $\overline{\text{SYNC}}$ pins high (effectively disabling them) when there is no desire to control these pins via software.



(a) Top view

(b) Bottom view showing isolated analogue ground (AGND) and digital ground (DGND) planes

Figure 6.11: 3D renders of the designed AD7730 breakout board with only surface mount components displayed

The motherboard

As discussed above, a motherboard was designed in order to connect all constituent subsystems of the data acquisition system to the micro-controller. Similar design principles as laid out in the design of the AD7730 board were followed as far as possible. It was decided that due to time constraints, pin sockets to house the STM32F4 Discovery board would be implemented on this board rather than designing a custom board around the STM32F4 MCU itself. The schematic for the motherboard from which the corresponding PCB was designed can be found in Appendix E.1. 3D renders of the motherboard are shown in figures 6.12 and 6.13 below

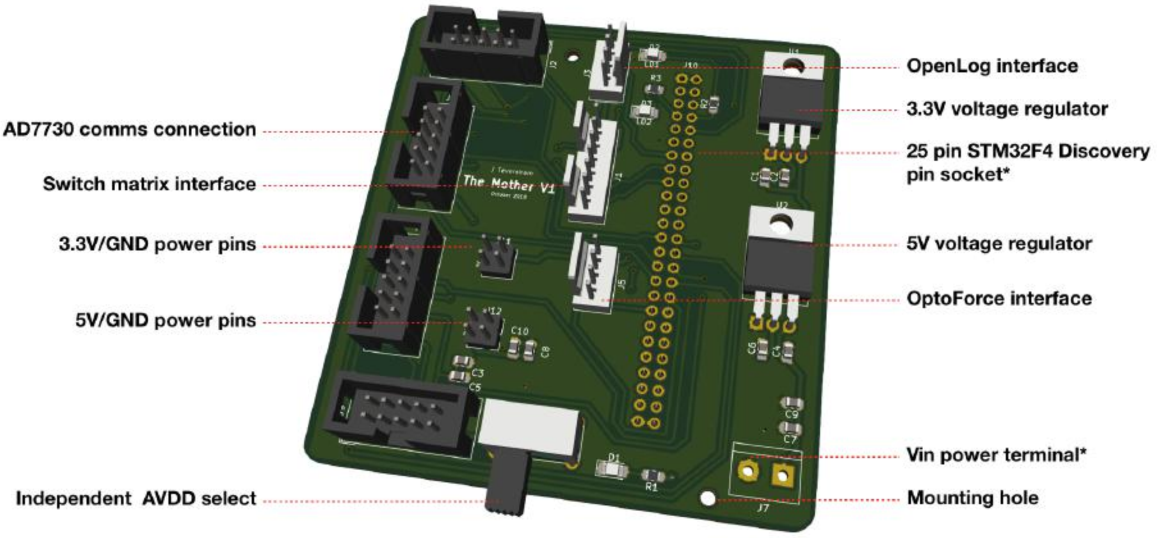
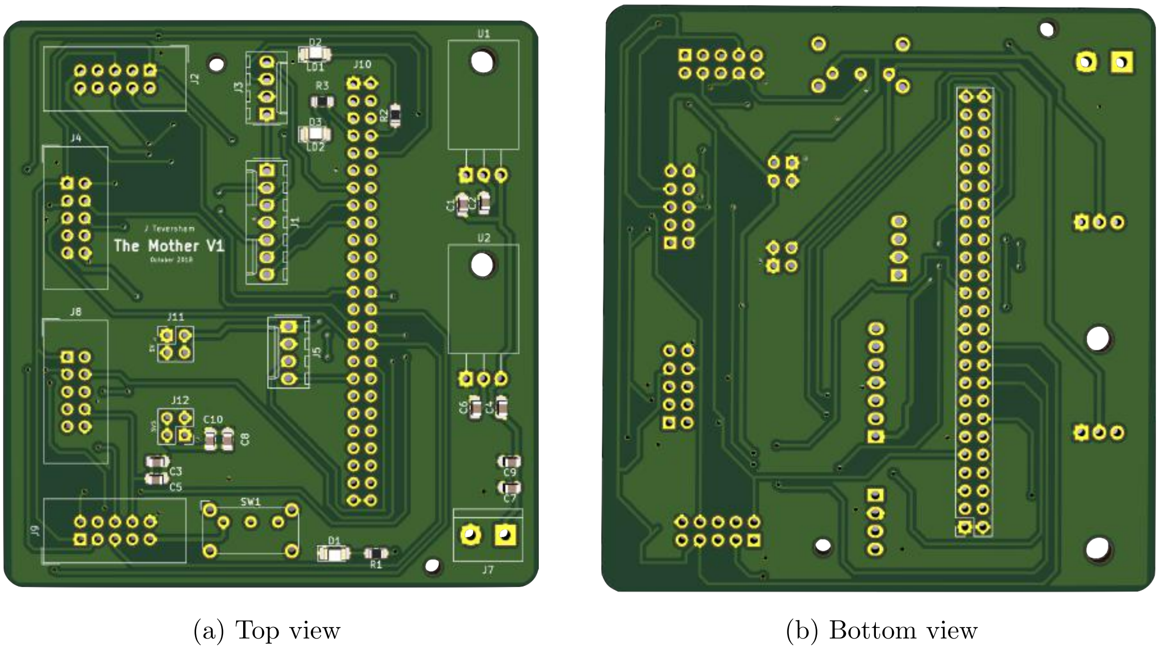


Figure 6.12: Annotated 3D render of the designed motherboard with surface mount and through-hole components displayed. The 24-pin micro-controller socket has been omitted to provide a better view.

Note that in the annotated diagram in Figure 6.12 above, labels with asterisks denote 3D render objects that were either omitted in order to prevent obscuring the other components or were not available in the KiCAD package. Furthermore, only one of the four IDC plugs is labelled 'AD7730 comms connection', however all four plugs serve this same purpose (for their respective chip). The 'independent AVDD select' switch was incorporated to allow the analogue power supply to the AD7730 chips to be independent (i.e 5V) or the same as the digital supply (3.3V) to the chips.



(a) Top view

(b) Bottom view

Figure 6.13: 3D renders of the designed AD7730 breakout board with surface mount components displayed only

Chapter 7

Software Design

This section details the design procedures followed in developing all embedded software systems used in this project; primarily for the data acquisition system. The entire embedded code base was developed in C using the open source Eclipse/GCC STM32 tool-chain with OpenOCD debugging. ST's HAL (Hardware Abstraction Layer) library was used instead of the older and now unsupported Standard Peripheral Library for several reasons. The HAL library is the only library officially supported and maintained by ST and is designed to work with the convenient STM32CubeMX application. The HAL library was also developed with the ability to port code between STM32 sub-families (F0, F1,...) as a priority.

7.1 The Challenge of High Speed Logging

One of the greatest challenges faced by the embedded software system in this project was to achieve the objective of 1kHz sampling. This entailed co-ordinating all processes to receive, validate, packetise and send data (these processes are hereinafter referred to as a 'data cycle') in under 1ms. As the STM32F4 MCU used was clocked at a core frequency of 168MHz, time taken for processing of data and numerical operations was not of major concern. The biggest speed limitation was imposed by serial communication between the various peripheral devices in the system. This included:

1. Receiving 16 bytes of data via UART from the OptoForce sensor
2. Receiving 3 bytes of data via SPI from the 24-bit data register of each of the four AD7730 ICs
3. Exchanging 10 bytes of data via SPI from the switch matrix
4. Transmitting 35 bytes of packaged data to the OpenLog SD logger via UART

Of these tasks, the fourth one (transmitting to the OpenLog) required the most significant amount of time (in the order of 500μs) to complete. However, this was only achieved after several performance adjustments were made as described below. The meticulous design procedure followed to ensure that all communication tasks could be completed within 1ms is documented below.

7.1.1 Memory Management and the DMA

Traditionally, instructions on the micro-controller are executed sequentially by the CPU as the program counter is incremented. Therefore, the implication of this synchronous operation is that if transmission of x bytes to a peripheral is initiated by the CPU, the CPU is tied up (occupied) for a given number of clock cycles until the transfer of x bytes is complete.

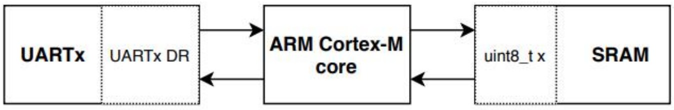


Figure 7.1: High level diagram depicting the CPU's involvement in a synchronous transfer of data between a variable in SRAM and the data register of a UART peripheral

This inefficiency can be circumvented to an extent through asynchronous operation offered by interrupts since peripherals can be programmed to generate an interrupt on several different events. For example, when receiving data via UART, the UART peripheral in use can be programmed to generate an interrupt with the RXNE (Rx data register not empty) flag set once the specified number of bytes is received [52]. Consequently, the CPU is free to perform other tasks until this interrupt is triggered and at such a time, can read the received bytes from the particular UART's data register. However, this approach still has the limitation that data must be moved between RAM and the peripheral registers by the CPU and interrupts need to be managed explicitly by the CPU.

The most efficient solution to this problem can be achieved through the use of the **DMA** (direct memory access) unit in the STM32F4. The DMA controller is a dedicated hardware unit (independent of the CPU) that allows peripherals in the MCU to access internal memories (RAM and flash) without intervention from the CPU core [52]. Apart from the small once-off overhead required to configure the DMA for transfers, the DMA controller operates independently of the CPU and can almost be considered as a memory co-processor. As a result, the CPU core is entirely free of the overhead created by data transfer as alluded to above [52].

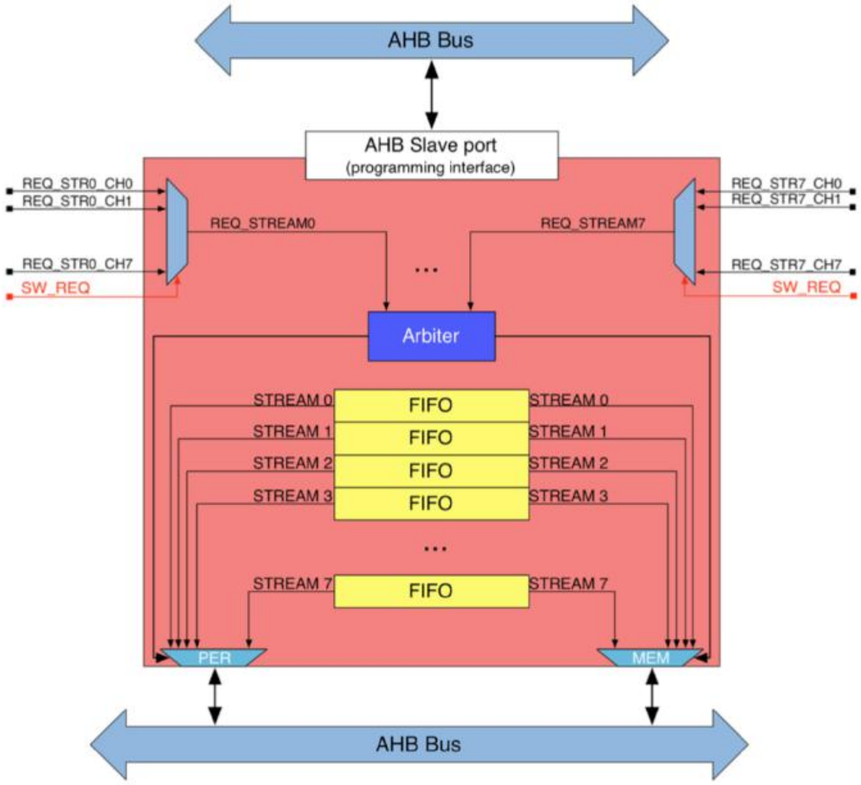


Figure 7.2: Diagram showing the DMA architecture in the STM32F4 MCU from [52]. The DMA controller is configured by the CPU via the AHB slave port interface.

The bus architecture of the STM32F4 MCU in Figure C.1 (Appendix C.1), shows how APB peripherals (such as the UART and SPI peripherals) can assert a DMA request to the DMA controller of either DMA 1 or DMA 2 which can then transfer data to/from that peripheral independently of the CPU core. Figure 7.2 above clarifies the DMA architecture in the STM32F4 MCU. It shows how groups of 8 DMA request channels are multiplexed to 8 independent streams dedicated to managing memory access requests from peripherals [52]. However, only one request channel per stream can be active at a given time and the DMA arbiter is responsible for managing priority of DMA requests [52]. This prevents contention between several peripherals requesting memory transfers using the DMA. Optional FIFO buffers can be enabled between peripherals and memory locations although if left disabled by default (as in this project), the configured peripheral and memory addresses become directly coupled during data transfer.

7.1.2 High Speed Logging with OpenLog

Initially, a baud rate of 115200 bps was used in transmitting data to the OpenLog logger as this was the maximum recommended rate documented on the SparkFun[1] OpenLog tutorial page. However, it was soon realised that transmitting 35 bytes at 115200 bps would require $t = \frac{\#\text{ bits}}{\text{bits/s}} = \frac{35 \times 8}{115200} = 2.43$ ms. In fact, this was experimentally shown on an oscilloscope

[1] SparkFun is the manufacturer of the OpenLog device

to be closer to 2.7ms (presumably due to the inclusion of start and stop bits in the UART transmission). Clearly, this was unacceptable as the full transfer of data to the SD card was required to take significantly less than 1ms in order to achieve 1kHz sampling[2]. Upon more thorough inspection of the OpenLog datasheet, it was noticed that baud rates of up to 2Mbps were physically supported by the MCU on board the OpenLog device (Atmel ATmega328p). However, the challenge remained to find a baud rate significantly higher than 115200 bps that could be matched *exactly* to that of the UART peripheral being used on the STM32F4.

In order to gain insight into their internal baud rate generation structures, the datasheets of the ATmega328p [53] and STM32F407 [54] were consulted. According to [53], the equation for calculating the UART baud rate is given by:

$$\text{baud} = \frac{f_{osc}}{8(\text{UBRRN}+1)} \text{ where UBRRN} = \frac{f_{osc}}{8 \times \text{baud}^*} - 1 \tag{7.1}$$

where baud* is the desired (theoretical) baud rate and baud (without *) is the actual baud rate used by the MCU. Because UBBRN must be an integer, these may not always be equal - if the combination of baud* and f_{osc} supplied yield a float (decimal), the decimal points will be truncated to produce a rounded down integer. This has the implication that for f_{osc}=16MHz (as with the OpenLog device), a finite number of baud rates are available: 2Mbps, 1Mbps, 666.67bps, 500000bps... for UBBRN = 0, 1, 2, 3... respectively. It was decided that a baud rate of 500 000bps was most suitable as it satisfied the speed requirements for the length of data to be transmitted but was not excessively high so as to increase the risk of transmission errors.

For the STM32F407, a greater resolution of exact baud rates is possible. According to [54], the equation used to calculate the UART baud rate for this MCU are is shown below. It should be noted that in order to gain the greatest resolution of possible baud rates, the STM32F4 was clocked at its maximum core frequency of 168MHz, resulting in the APB2 bus connected to USART2 (used for the OpenLog device) being clocked at its maximum of 42MHz.

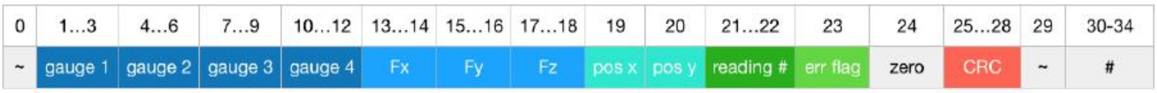
$$\text{baud} = \frac{f_{ck}}{8(2 - \text{OVER8}) \times \text{USARTDIV}} \tag{7.2}$$

where OVER8 is a logical variable that is 0 when UART oversampling by 16 is used as in this system. USARTDIV is an unsigned fixed point number coded to the USART_BRR register [54]. It is composed of an 8-bit mantissa and 4-bit fractional component. To achieve a baud rate of 500 000bps for this MCU, USARTDIV = $\frac{42e6}{8 \times 2 \times 500e3}$ = 5.25. This results in a mantissa of 5 and fractional component of 0.25 = $\frac{4}{16}$. The resulting value coded to the USART baud rate register was thus USART_BRR = 0x054. This allowed the baud rate of *exactly* 500 000 bps to be achieved by both MCUs and resulted in extremely consistent logging performance.

[2] sampling here refers to the whole process of gathering data, packetising and logging - a full data cycle

7.2 Data Packetisation

It was realised that optimising the process of logging to the SD card for speed was more important than having a more readable log format such as when logging with ASCII characters and delimiters. As such, raw bytes were logged to the SD card using a predefined packetisation scheme with a data packet structure as depicted in Figure 7.3 below. A tilde (hex 0x7E, decimal 126) was used as the start and stop character (byte).



0	1…3	4…6	7…9	10…12	13…14	15…16	17…18	19	20	21…22	23	24	25…28	29	30-34
~	gauge 1	gauge 2	gauge 3	gauge 4	Fx	Fy	Fz	pos x	pos y	reading #	err flag	zero	CRC	~	#

Figure 7.3: Diagram showing packet structure used to transmit data to the micro SD card via UART. The top row shows byte positions and the bottom row, the data assigned to the corresponding bytes.

The descriptions of the data assigned to the various bits in Figure 7.3 are clarified below:

- **gauge** i: the three data bytes from the 24-bit data register of the ith AD7730/load cell combination
- **Fx, Fy, Fz**: the three 16-bit GRF component forces captured by the OptoForce sensor
- **pos x, y**: one byte for the x co-ordinate of the detected COP position and byte for the y co-ordinate
- **reading #**: the reading number recorded by the OF sensor used as a reference
- **err flag**: error flag byte used to indicate the presence of sampling errors that can be handled during data post-processing[3]
- **zero**: zero byte used for padding to make the data packet length word (32-bit) divisible (explained in Section 7.3.1 below)
- **CRC**: 32-bit CRC word used for verifying data transmission integrity during post-processing

Using a tilde as the start and stop (terminal) character led to a limitation: if another byte within the data packet (bytes 1 to 23 in Figure 7.3) had the decimal value 126, the data decoding program would incorrectly interpret it as an early stop character. Furthermore simply interpreting the stop character as a fixed length of bytes after the start character proved ineffective in practice as occasionally (albeit very rarely), a byte was dropped during transmission and ruined the sequence. The solution implemented was to predictably change any non-terminal 0x7E bytes to a sequence of two known substitute bytes which could then be recognised and restored during decoding of the logged data.

[3] decoding and post-processing of logged data in MATLAB

7.3 Data Integrity Measures

7.3.1 Cyclic Redundancy Check (CRC)

CRC codes are error-detecting mechanisms that are commonly used to verify the integrity of transmitted data. CRCs are significantly more robust and effective that checksums and parity bits (which are essentially 1-bit CRCs) and are ideal for the application in this project as they are particularly effective[4] at detecting errors caused by noise along transmission channels and are simple to implement in binary hardware [56]. A particular generator polynomial is required in the system-specific CRC calculation which becomes the divisor in a polynomial long division operation [56]. The input data forms the dividend, the quotient is discarded and the *remainder* is the CRC code result.

In order to verify that no corruption occurred during transmission of packetised data to the micro SD card, the following process was followed: for each data packet, a 32-bit CRC code was first calculated by the hardware CRC unit of the STM32F4 - an impressive 60 times faster than the same algorithm implemented in software with level 3 optimisation [55]. As shown in Figure 7.3, the CRC code was appended to the packet before transmission (note the start and stop bytes did not form part of the data used to calculate the CRC code). However, since the hardware CRC unit in the STM32F4 only operates on 32-bit word arrays, a zero byte was appended to each data packet (as seen in Figure 7.3) in order to ensure that the data packet was word-aligned[5]. The algorithm used by the CRC hardware unit in the STM32F4 is clarified by the flowchart in Figure 7.4 below. This flowchart was used to develop a MATLAB code implementation that was used to recalculate a CRC code based on the received data. This was then compared with the MCU-calculated CRC in the received data packet to verify the integrity of the received data. Further details of the MATLAB decoding process are discussed in Section 8.1.1.

[4] In fact, the STM32 CRC implementation is used in flash content integrity self-test checks in ST firmware [55]

[5] i.e. able to be split into an array of 32-bit words

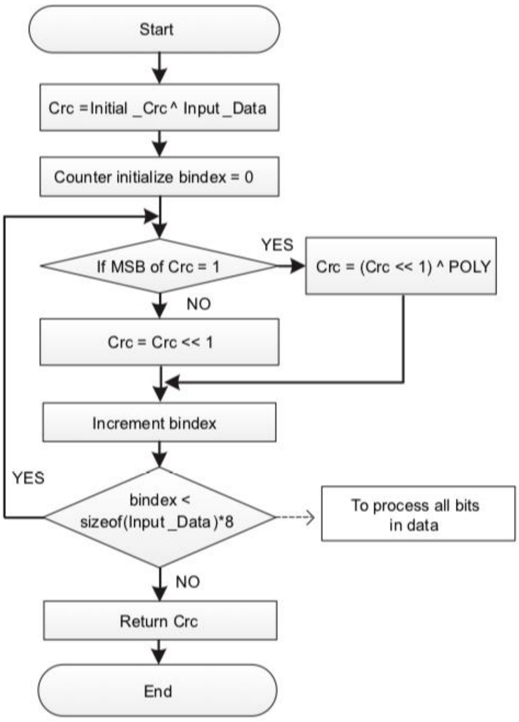

Start
Crc =Initial _Crc ^ Input _Data
Counter initialize bindex = 0
If MSB of Crc = 1
YES
Crc = (Crc << 1) ^ POLY
NO
Crc = Crc << 1
Increment bindex
YES
bindex < sizeof(Input _Data)*8
To process all bits in data
NO
Return Crc
End

Figure 7.4: Flowchart extracted from [55] representing the CRC calculation algortithm used by the STM32Fxx MCU series

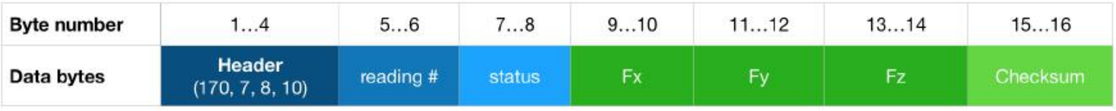
Note that the initial CRC and generator polynomial (Initial_Crc and POLY respectively in Figure 7.4) are fixed in the STM32F4. Their values are 0xFFFF FFFFF and 0x4C11DB7 (the CRC-32 Ethernet polynomial) respectively.

7.3.2 OptoForce Packet Checking

The packet structure used by the OptoForce sensor is depicted in Figure 7.5 below. As the 3 axis GRF components measured by the OptoForce sensor were used as the truth data (targets) for the machine learning models in this project, it was of paramount importance that the integrity of this received data could be verified. To this end and with reference to Figure 7.5, the following checks were implemented on each 16 byte data packet received from the OF sensor:

- The four **header bytes** were checked to match the constant pattern of (170,0,50,3) as in every valid packet
- It was verified that the **error bytes** were both zero indicating no data conversion or other internal sensor errors occurred
- The **checksum** was checked to match the sum of the rest of the data bytes in the packet

While still logged, any invalid packets (that failed any of the above checks) were flagged by setting the OptoForce error bit in the error flag byte as per Figure 7.3.



Byte number	1...4	5...6	7...8	9...10	11...12	13...14	15...16
Data bytes	**Header** (170, 7, 8, 10)	reading #	status	Fx	Fy	Fz	Checksum

Figure 7.5: Diagram to clarify the structure of the 16 byte packets transmitted by the OptoForce sensor

7.4 AD7730 SPI Interface Library

As mentioned in Section 6.2.1, very little information pertaining to interfacing with the AD7730 (besides the technical data presented in the datasheet) was available across the internet and in other resources. As a result, a comprehensive AD7730 SPI library[6] written in C was developed for interfacing with the STM32F4 MCU. This section will explore some of the principles followed the design of this implementation.

7.4.1 Register Structure and Read/Write Procedure

Vital to the understanding of the operation of the AD7730 is the on-chip register structure depicted in Figure 7.6 below. With the exception of continuous read operations as described below, all transactions with on-chip registers must begin with a write to the communications register. This initial write command specifies both whether the next operation is a write/read command, and the target register for the subsequent operation [49]. A subsequent command is sent to either read or write to the previously specified target register (such as a write to the filter register to configure the chip's filter settings) and following this, the chip returns to an idle state (where it began) where it expects another write to its communication register.

As shown in Figure C.2, two predefined bits (RW1 and RW0) are used to selected the type of read/write operation and three other bits (RS0 to RS2) are used to select the target register. The particular bit patterns corresponding to these bits are also presented in Figure C.2. Note that all these bits form part of the communications register. Flowcharts extracted from [49] that clearly explain the procedure required for reading from and writing to the AD7730 (in all modes) are presented in Appendix C.2.

[6] available at: `https://github.com/JamesTev/AD7730`

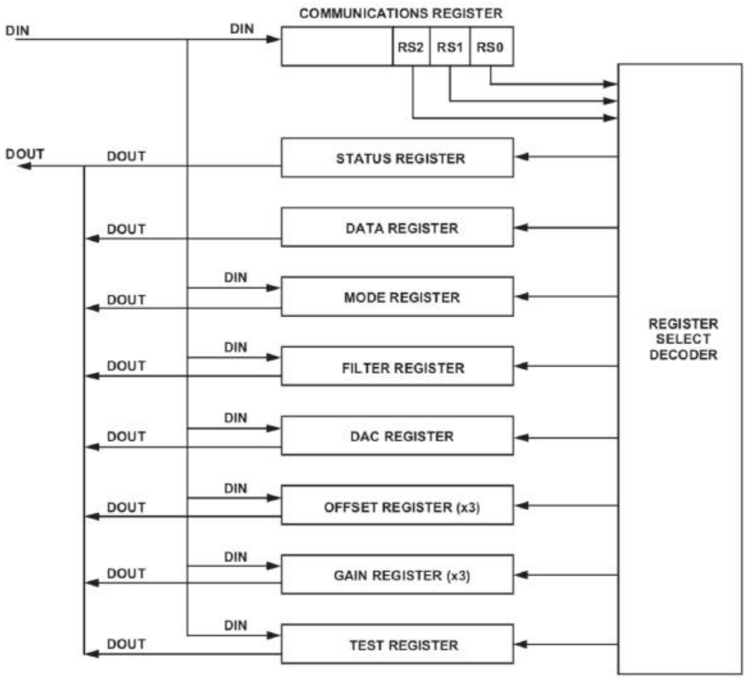


Figure 7.6: Diagram from [49] showing the on-chip register structure. Note that the communications register directly controls (provides input to) the register select decoder.

Continuous read mode

In continuous read mode, the process of interacting with the AD7730 is slightly different. Once programmed to start continuous read, the three bytes of the 24-bit data register of the chip can be read *without* any intermediate commands to the communications register as was previously required above. The $\overline{\text{RDY}}$ pin goes low once a conversion is complete and thus can be monitored via hardware (GPIO pin on the MCU) to determine when the data register can be read. The $\overline{\text{RDY}}$ pin returns high once the data register has been read. Once appropriately configured, the AD7730 chips were used in continuous read mode in this project in order to reduce communication overhead and facilitate operating at an ODR of 1kHz.

Filter configuration and ODR control

The three bytes comprising the filter register on board the AD7330 not only control the filter characteristics and operation, but also directly control the ODR. This is because the amount of averaging (effectively the extent of down-sampling) performed by the sinc^3 filter in the AD7730 influences the final ODR. As documented in Figure C.2, 12 bits (SF0 to SF11) control the ODR according to the following formula[7] [49]:

$$ODR = \frac{f_{CLK(in)}}{16} \times \frac{1}{SF} \tag{7.3}$$

[7] this assumes input chopping is disabled which is required in order to reach an ODR of 1kHz

where SF is decimal equivalent of the data provided in the SF bits of the Filter Register and $f_{CLK(in)}$ is the master clock frequency (of the external crystal) [49]. In order to achieve an ODR of 1kHZ, the SF bits were set to $SF = 312$ (decimal) to yield a theoretical ODR of 1001 Hz[8].

SPI multi-slave structure

Figure 7.7 below clarifies the structure used to communicate with the four AD7730 chips independently on common SPI lines. In order to communicate with the ith chip, only the ith SS (slave select) line was pulled low and the rest were kept high.

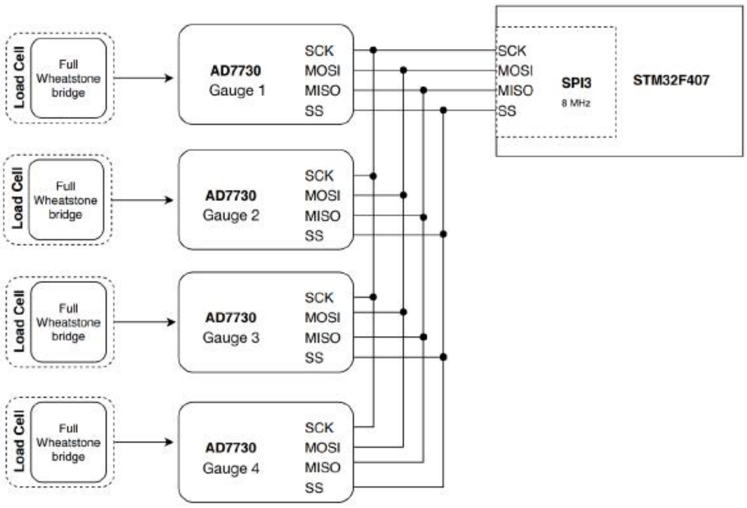


Figure 7.7: Diagram showing the master (STM32F4 MCU) and multi-slave SPI structure used to communicated with the four AD7730 chips independently with common SPI lines.

7.5 Switch Matrix

7.5.1 Serial Interface

In order to interact with the switch matrix shift register hardware as discussed in Section 6.2.2, an SPI interfacing scheme was developed. In order to read switch presses unambiguously from the matrix, only one row was set high (enabled) at any given time. In order to enable row i, a byte with only the ith bit set was transmitted to the matrix (specifically to the serial-in line of the 74HC595 SIPO shift register) via SPI. A byte representing the logic state of the matrix columns was then received (shifted out) from the matrix via the 74HC165 PISO shift register. By observing this received byte, it could be deduced which switches in row i were pressed due to

[8] The AD7730 datasheet [49] recommends using a 4.9125MHz crystal in which case an ODR of exactly 1kHz is attainable. However, such a crystal was not readily available and a 5MHz crystal was also listed as acceptable in [49].

the direct correspondence of set bits in this byte and switch positions for the ith row. Although not all switches could strictly be read from *simultaneously*, reading from the matrix with 5 rows required a (bidirectional) transfer of only 10 bytes (and in general, $2n$ bytes for n rows[9]). Because SPI1 in the STM32F4 was configured to run at over 4MHz, several thousand checks of the full matrix could be performed per second.

7.5.2 Positional Interpolation

Due to the fact that a discrete grid of 26 switches was used to capture the COP position from a continuous xy plane, a means of detecting positions in between discrete grid points needed to be devised. In order to achieve this, the centroid - as defined by Equation 7.4 below - of all point co-ordinates (x_i, y_i) corresponding to N triggered switches was computed and used as the interpolated position.

$$(x_c, y_c) = \frac{1}{N}(\sum_{i=1}^{N} x_i, \sum_{i=1}^{N} y_i) \tag{7.4}$$

where (x_c, y_c) is the co-ordinate of the geometric centroid of a set of N points in an xy plane.

7.6 Combined Embedded System

The flowchart in Figure 7.8 below clarifies the data acquisition process. Importantly, the process blocks with dotted green borders represent asynchronous tasks handled by the DMA. Since the SPI communication with the AD7730 chips and the switch matrix could be run at over 4MHz, it was decided to simply use the CPU to manage these processes. Furthermore, if both the SPI transfers and the UART transfer from the OF sensor were delegated to the DMA, the CPU would be left completely unoccupied but would have no other tasks to execute anyway until all data was received from these peripherals. Although the CPU would physically be free to resume sequential execution once the data from the switches was received, the data would not be able to be fully processed and packetised until the full 16 bytes of data from the OF sensor (handled by the DMA) were received.

[9] with a maximum of 8 rows and 8 columns

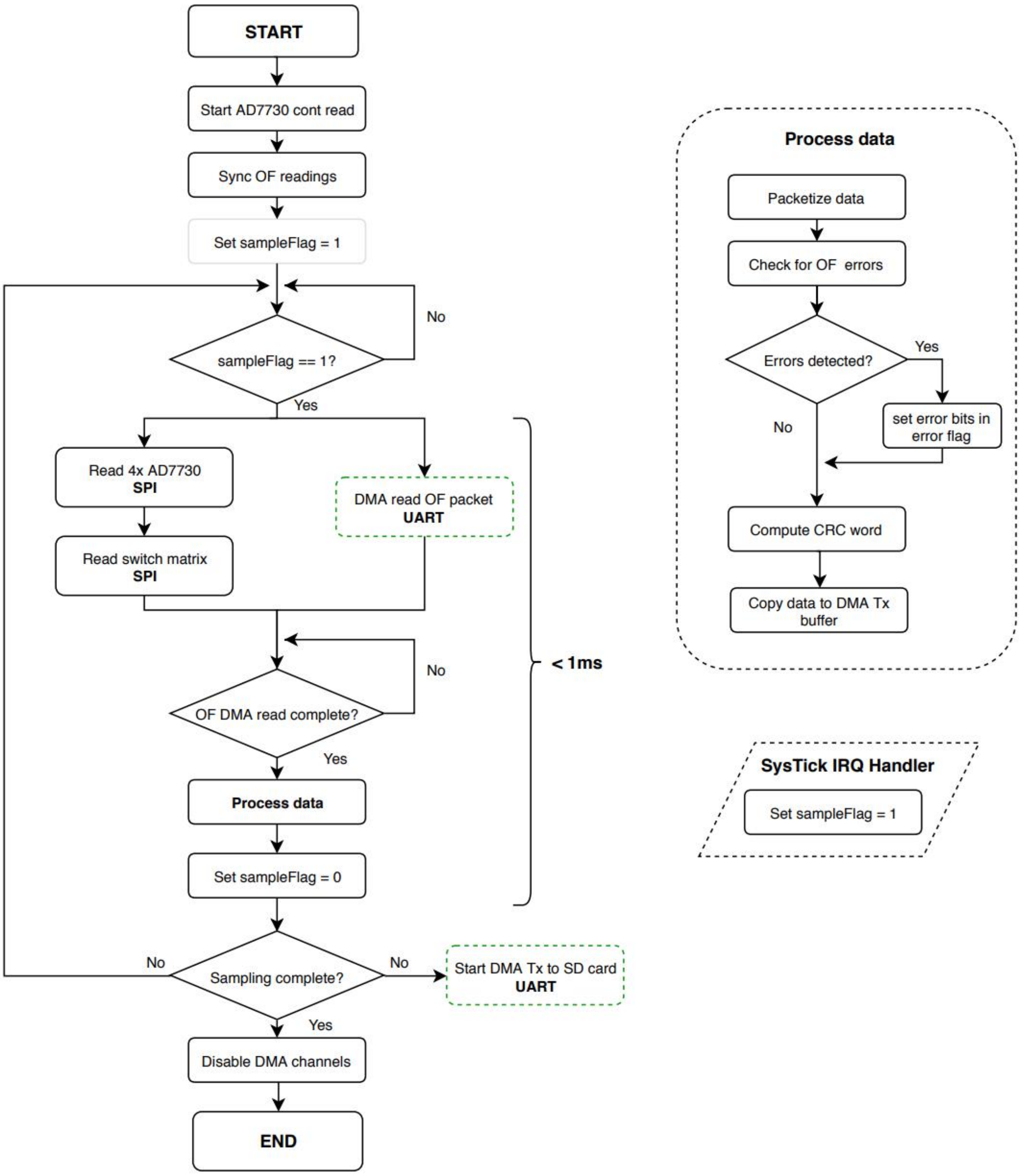


Figure 7.8: Flowchart of the data acquisition process co-ordinated by the STM32F4 MCU. The *Process data* block is expanded alongside. The SysTick IRQ Handler is a parallelogram to represent an independent asynchronous event.

7.6.1 Significance of Leveraging the DMA for Data Logging

Another important point to note is that upon completion of packetisation of data, only a transfer request needed to be sent to the DMA and after doing so, the CPU could return to polling the sample flag for the next data cycle. This is subtlety is important; the only processes in the flowchart that needed to necessarily complete in under 1ms (hereinafter referred to as the *critical data cycle*) were those indicated by the large bracket in Figure 7.8. If this was achieved, the UART transfer to the SD card managed by the DMA had *at least* 1ms to complete since the transfer could be executed in parallel to the CPU and once initiated, only needed to complete before a new transfer was initiated at the *end* of the following *critical data cycle*. If the DMA was not used for this purpose, the CPU would be required to handle the transfer and would only have 1ms less the duration of the *critical data cycle* to do so. From experiments performed using the oscilloscope, the length of the data cycle was typically in the region of 700μs - leaving only 300μs for transmission of a full data packet to the SD card (if the CPU were to be used).

Chapter 8

Machine Learning Implementation

A core component of this project was the use of machine learning (ML) techniques to learn non-linear force relationships presented by the mechanical design of the system. This learning process was performed carefully and systematically through the training of various ML models in a variety of tests described below. This brief chapter discusses data post-processing techniques used and outlines the software implementations of some of the machine learning models detailed in the Theory Development in Chapter 3.

Figure 8.1 below outlines the basic input-output structure used in machine learning regression models in this project; G_i for $i \in [1:4]$ represents the force reading input from gauge[1] i and inputs x and y encode the co-ordinates of the COP detected by the switch matrix. F_x, F_y and F_z are the output[2] GRF component variables.

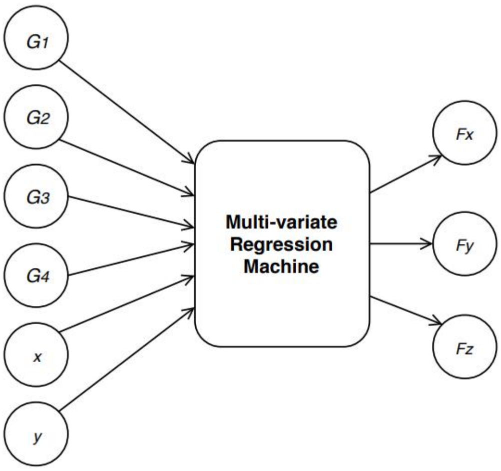


Figure 8.1: Diagram to show the input-output structure of various regression models used in this project

[1] gauge i, as loosely referred to commonly in this report, actually refers to load cell i

[2] the output parameters represent the actual responses from the OF sensor during training and estimated responses during testing

8.1 Software Implementation

MATLAB was used almost exclusively for the implementation of machine learning models in this project since it offers a stand-alone[3] platform, excellent built-in functionality for quick testing and convenient data-visualisation tools. It is also extremely well optimised for processing large volumes of data.

8.1.1 Decoding of Log Data in MATLAB

On every power cycle of the OpenLog device, a new text file was created with an incremental file name such as ”LOG001.TXT”, ”LOG002.TXT” etc. The challenge remained to read in and interpret the raw data in each text file log byte-by-byte. MATLAB was used for this purpose - specifically, the MATLAB `fread` with `uint8` argument was used to read in individual bytes. The same algorithm used in Section 7.2 for dealing with non-terminal tilde (0x7E) characters was used in reverse to restore each data packet to its original form (as created on the MCU, before transmission). Finally, for each packet, the CRC code was calculated and compared with the CRC code received in the data packet. If these did not match or any bits in the error flag byte (as explained in Section 7.2) were set, the whole packet was flagged and excluded from the final array of processed data packets. Because this occurred so rarely, it was deemed the most acceptable option to merely discard erroneous packets. Finally, with knowledge of the packet structure outlined in Figure 7.3, the array of processed data packets was read into variables in MATLAB using suitable typecasting and saved into .mat files for fast and convenient processing at a later stage. The extraction script (`analyse_data_V2.m`) used to process raw log data can be found at: https://github.com/JamesTev/Force_Plate/tree/master/MATLAB%20Code/FP_Utils.

8.1.2 Data Normalisation

It was noticed that while all load cell readings had a fixed and consistent zero point which was measured when no force was applied, this point varied across the readings by a constant offset (termed the ‘zero level’). Because the zero point was particularly consistent in each data set, no force was applied for approximately the first second of logging. This allowed at least 500 zero points to be captured and the fixed zero level of each sensor to be determined. This offset was then subtracted (or added for negative offsets) from the data of each load cell to produce consistent, zero-mean readings. Finally, the signed 24-bit ADC-converted load cell readings were normalised to a percentage of the full scale reading by dividing by $(2^{23} - 1) = 8388607$ (maximum signed 24-bit number in decimal). Figure 8.2 below shows a sample of normalised data. It should be noted that the zero level was independent of the magnitude of each gauge reading - these readings are unequal due to a random force applied at a non-central position on the plate.

[3] no need for installation of external libraries or environments

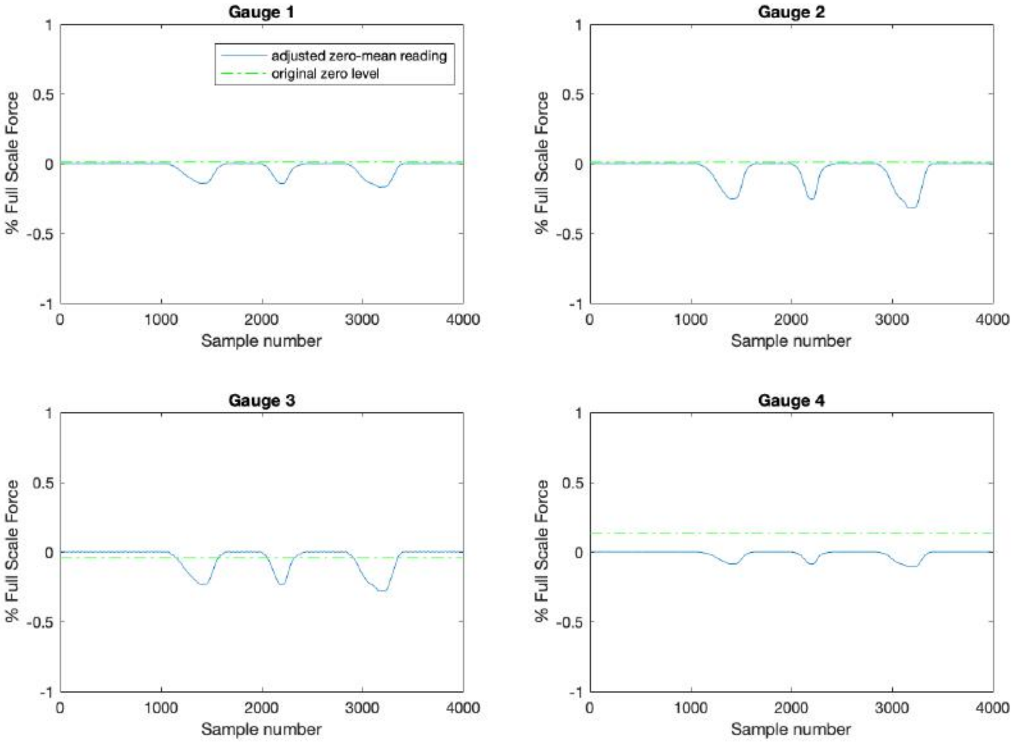


Figure 8.2: Diagram to show adjusted zero-mean load cell ('gauge') readings for an arbitrary force profile. The legend applies to all sub-figures.

8.1.3 Linear and SVM Regression

Linear regression, as described in Section 3.1.1, was implemented using MATLAB's `fitrlinear` function. This function accepted a $[n \times p]$ training input matrix `X_train` (n observations and p features or regressors) and a $[n \times 1]$ target vector `Y_train`. Unfortunately, only uni-variate (single output) linear regression models were available in the version of MATLAB used (R2016b). As such, the structure shown in Figure 8.1 had to be adapted to instead train three separate linear models, each with one of the 3D GRF vectors as the output variable. However, each model was presented the same input data - only the outputs were decoupled.

Epsilon-insensitive SVM regression, as described in Section 3.1.1, was implemented using MATLAB's `fitrsvm` function. The type of SVM kernel used was simply altered using the `'KernelFunction'` parameter. In this project, only linear and Gaussian kernels were experimented with as the polynomial kernel model often required inordinate training time in comparison. As with the linear model, only uni-variate (single output) SVM models were available in the version of MATLAB used. The same strategy of three independent, output-decoupled models as used above was implemented.

8.1.4 Feed-forward Neural Network

MATLAB's Neural Network Toolbox was largely used for creating and training of neural nets in this project. This toolbox offers a powerful implementation of a single-hidden-layer feed-forward network with a flexible architecture; the number of input and output neurons are determined by the dimensions of the input and output data matrices respectively and the number of neurons in the hidden layer can be controlled directly. The network, by default, uses a non-linear (sigmoidal) activation function between hidden and output layers and a linear function on the output layer (as recommended in the Theory Development in Section 3.1). Furthermore, the training algorithm used in tuning the network parameters can be selected from three algorithms, namely: the Levenberg-Marquadt, Bayesian regularisation and Scaled Conjugate Descent algorithms. Except for any explicit mentions to the contrary in specific experiments in this project, the default training algorithm used was the Levenberg-Marquadt algorithm due to its fast training time and consistently robust performance (as determined experimentally). Networks trained with Bayesian Regularisation yielded almost indistinguishable results in each experiment.

Data division

During training of neural networks using this tool box on a particular data set, data (including network inputs *and* output targets) was randomly divided using `dividerand` into 3 subsets of predefined proportions:

- **Training**: data presented to the network during training - used to adjust the network parameters based on evaluation of the cost function and other parameters. 70% of the whole data set was typically allocated to training.
- **Validation**: data samples used to measure the network generalisation and to stop training when generalisation stopped improving. During validation, out-of-sample (unseen) data would be presented to the network and the subsequent estimation performance would yield validation performance (and thus, an indicator of generalisation). 15% of the whole data was typically allocated to validation.
- **Testing**: data samples used to independently[4] evaluate network performance during and after training. 15% of the whole data was typically allocated to testing.

Network parameters

One of the difficult parameters to optimise in a neural network is the number of neurons in the hidden layer. As discussed in Section 3.2.5, a network with a larger number of hidden neurons

[4] testing with this subset has no effect on training

has increased predictive power but consequently, is more susceptible to over-fitting training data (and poor out-of-sample performance) if sufficient regularisation is not implemented. Thus, the challenge remained to determine an optimal number of hidden neurons to achieve the best balance between predictive power and generalisation to out-of-sample data. Due to the unknown, non-linear relationships represented by the network, this parameter was one that had to be optimised empirically. Although this could not be done as efficiently as with the ELM described below, performance of networks with varying numbers of hidden neurons (typically between 5 and 20) was evaluated for each experiment. The **default** number of hidden neurons used in experiments (unless otherwise specified) was 10. While the learning rate (as referred to in Section 3.2.4) was adaptively adjusted with the Levenberg-Marquardt algorithm, the MATLAB NN Toolbox did not allow for modification of the cost function to introduce λ-penalisation for direct control of generalisation (as outlined in Section 3.2.5).

8.1.5 ELM

Due to the relative ease of implementing a basic ELM in MATLAB, as well as the fact that MATLAB does not offer a built-in ELM model, a code implementation using singular value decomposition and basic linear algebra techniques was developed. The basic idea behind this implementation for `n` hidden neurons is shown in the listing below:

```
Xa = [ones(length(X), 1) X]; %append column of ones to input matrix
R = 2*randn(size(X, 2)+1, n); %uniformly distributed random numbers in [-1;1]
H = tanh(Xa*R); %create design matrix with hyperbolic tangent activation fn.
[U, S, V] = svd(Xa, 'econ'); %orthogonal matrices from singular value ...
    decomposition
P_inv = V*inv(S+lambda*eye(length(S))*U'); %compute pseudo-inverse with ...
    regularisation
w = P_inv*Y;

Qa = [ones(length(Q), 1) Q]; %augmented query matrix
Y_pred = Qa*w; %predicted targets (testing)
```

The matrices `X` and `Y` contain the training data inputs and targets respectively. Matrix `Q` represents a matrix of out-of-sample test inputs and `Y_pred` is the corresponding matrix of predicted test outputs. As explained in Section 3.3, the ridge factor, referred to as `lambda` above, is responsible for regularisation and is also determined empirically. However, as discussed in Section 3.3, for each re-evaluation of generalisation performance with a different values for `lambda`, only matrix multiplication is required (no matrix inversions) and a wide range of `lambda` values can be tested extremely quickly and efficiently, even for networks with in excess of 200 hidden neurons. Algorithms to perform efficient numerical optimisation of the ridge factor λ and the number of hidden neurons in the ELM models developed are presented in the project code listing available at: `https://github.com/JamesTev/Force_Plate/tree/master/MATLAB%20Code`.

8.1.6 Other Implementations

Apart from the MATLAB regression models mentioned above, many different implementations exist on various platforms. Keras is a particularly popular open source neural network library written in Python. It serves as a high-level API that runs on top of powerful machine learning frameworks such as Google's TensorFlow and Theano and allows for greater flexibility in designing the architecture of the neural network. Experimentation with a SFLN neural network with varying numbers of hidden neurons was performed using Keras and a TensorFlow backend. As with MATLAB models, a MSE cost function was used. However, various tests conducted with a range of different network parameters yielded almost identical results to comparable MATLAB models. The Keras model implemented in Python is available at: `https://github.com/JamesTev/Force_Plate/tree/master/Keras%20Model`.

Chapter 9

Results

This section presents the results obtained from simulations and practical experiments conducted. Note that in most graphs shown in this section, axis labels F_x, F_y and F_z represent the component forces in the x (mediolateral), y (anterior-posterior) and z (vertical) axes respectively. Furthermore, error metrics are often presented as a percentage of the full scale for (%FS) which, as used in the calculations of this system, is 300N for the two lateral axes and 2000N for the vertical axis. RMS error, as used extensively in this section, is defined by Equation 9.1 below:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{9.1}$$

where N is the number of observations in the data set and y_i and $\hat{y}_i$ are the actual and estimated responses respectively for observation i. Note that unless explicitly stated otherwise, all results shown below show model performance on completely *unseen* data. Additionally, all co-ordinates are referenced from the plate co-ordinate scheme established in Figure A.1.

9.1 Preliminary Signal Tests

As mentioned in Section 8.1.2, load cell readings from the AD7730 chips showed extremely consistent zero levels with remarkably low noise. Figure 9.1 shows 1000 sampled zero readings (measured when no force was applied to the plate) with no software filtering or post-processing besides normalising each ‘gauge’ (load cell) reading as described in Section 8.1.2.

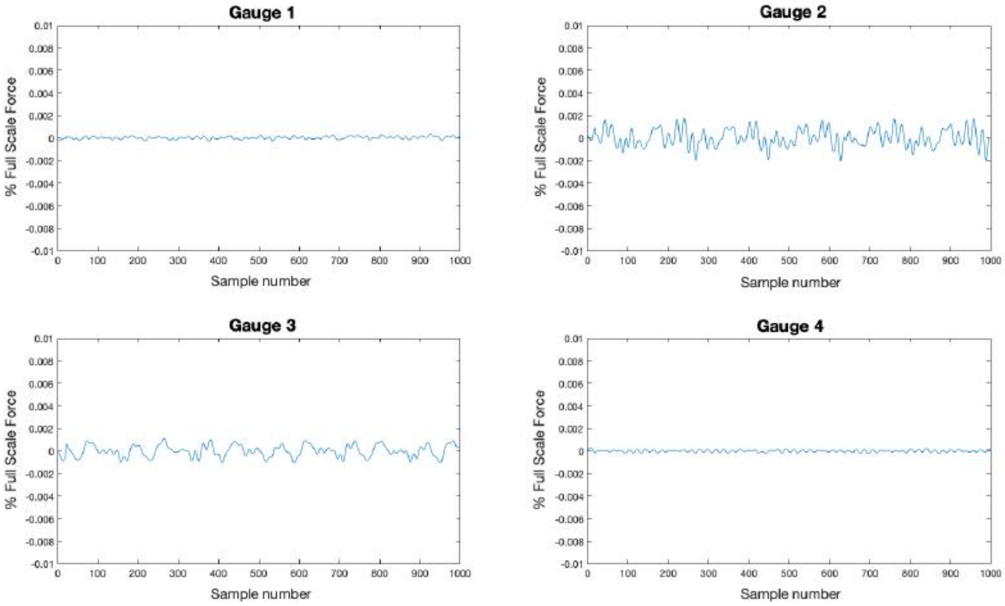


Figure 9.1: Diagram showing 1000 zero readings from the four load cells sampled at 1kHz. This data was captured using the prototype board shown in Figure 6.7. Notice the extremely small y-axis scale.

9.2 Single Axis GRF Estimation

As described in Section 4.2.2, training data was first obtained for the application of an exclusively vertical (z-axis) force. Figure 9.2 below compares the raw data obtained from the four load cells against the z-axis force (hereinafter referred to as F_z) measured by the OptoForce sensor for two different COP positions on the plate.

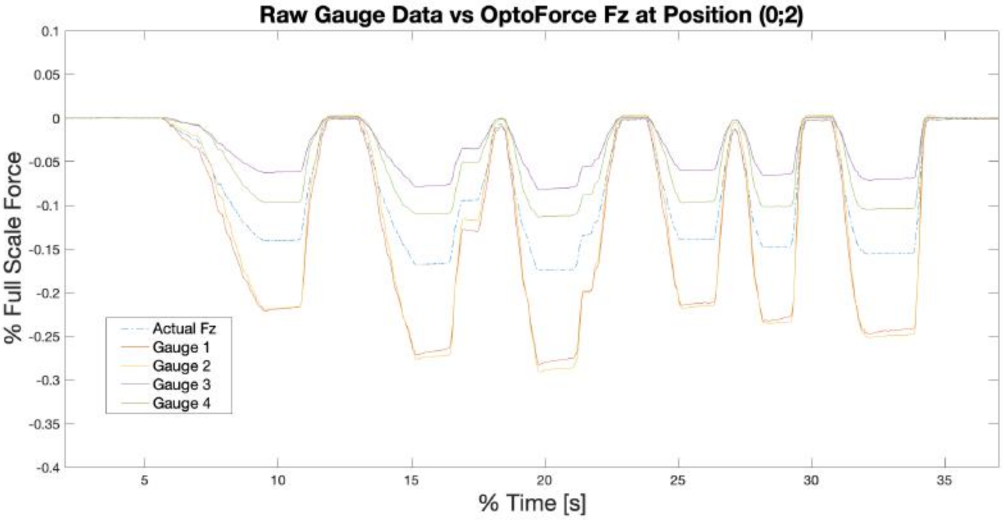


Figure 9.2: Graphs showing raw load cell data vs the actual F_z at a COP position of (0;2). The y-axis shows %FS of the full scale vertical force.

The difference in relative magnitudes of the four gauges in Figure 9.2 above is expected as the load was applied closer to load cells 1 and 2 at point (0;2) in this experiment. Furthermore, the gains of the load cell amplifiers (on board the AD7730 chips) were initially set to an arbitrary value (albeit the same value across the four amplifiers) in this experiment and thus the the magnitudes of the load cell ('gauge') readings relative to the OptoForce F_z measurement are not relevant.

9.2.1 Static Load Testing

Once it was established that the data acquisition system was providing reliable data, the simplest conceivable test of the regression models was the task of estimating a static vertical force induced by a fixed mass. This test was also used to serve as an indicator of the precision[1] of the measurement system (both the OptoForce sensor and the regression models). A neural network[2] with 8 hidden neurons and a linear regression model are compared in this simple test.

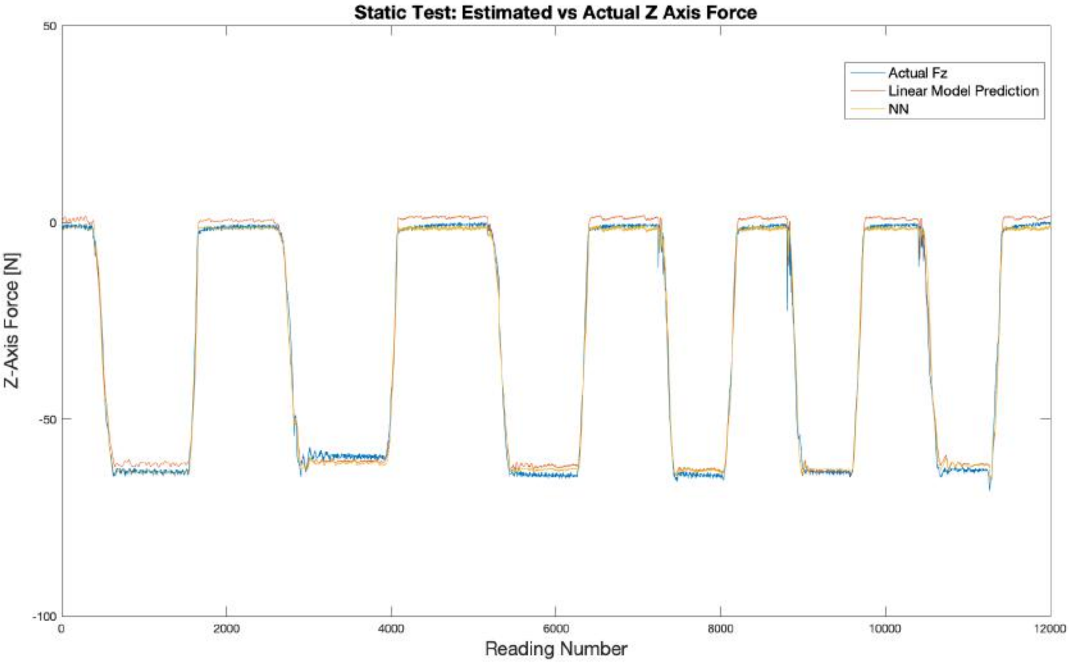


Figure 9.3: Graph showing the performance of two different models in estimating the force induced by a 6kg fixed mass.

9.2.2 Dynamic Loading

Following the static test performed above, an arbitrary dynamic force was applied using the drill press apparatus described in Section 4.2.2. Figure 9.4 below shows the estimated force response of various regression models at an arbitrary point (-2;-2). Table 9.1 below shows the

[1] the degree of similarity or repeatability between successive readings that ought to be identical

[2] unless otherwise specified, this refers to a net created with the MATLAB Neural Network toolbox with the default architecture discussed in Section 8.1.4

RMSE, standard deviation of absolute[3] prediction error and maximum (instantaneous) absolute prediction errors for the three regression models in the experiment.

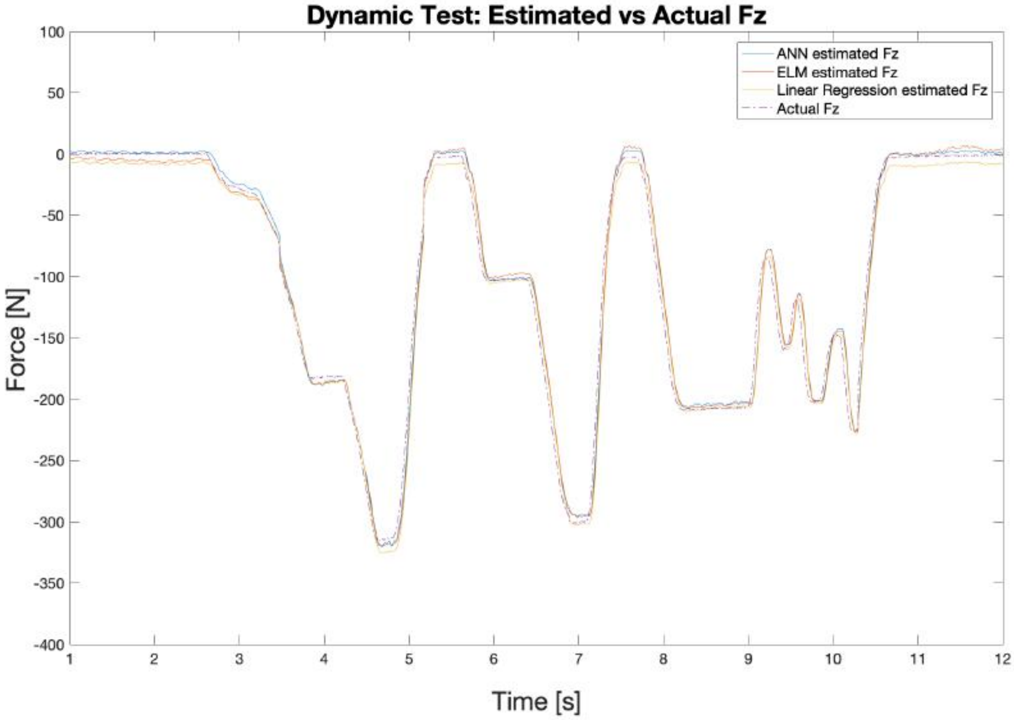


Figure 9.4: Graph showing the performance of three different models in estimating an arbitrary dynamic force in the z-axis applied at (-2; -2)

Table 9.1: Table displaying RMS error (RMSE), standard deviation of absolute error and maximum absolute error for each regression model presented in Figure 9.4

	RMSE		**SD of Abs. Error**		**Max Abs. Error**	
	[N]	% FS	[N]	% FS	[N]	%FS
Linear	11.157	0.560	15.892	0.795	42.920	2.146
NN	9.533	0.477	13.576	0.679	35.264	1.763
ELM	10.578	0.529	14.765	0.738	36.209	1.810

9.2.3 Positional Testing

Once the platform's ability to reliably estimate vertical forces applied at the *centre* of the plate was established, an experiment was conducted to determine whether vertical forces could be estimated with similar accuracy at any of the 26 designated points on the plate. As explained in Section 4.2.2, data was thus collected at each of the 26 switch positions depicted in Figure A.1. In Table A.1, the results for testing on unseen data on each of the 26 points are documented.

[3] calculated on the *magnitude* of errors between predicted and actual outputs

The RMS error, standard deviation of absolute error and maximum absolute error are presented both in units of newtons and more importantly, as a percentage of the full scale vertical force of 2000N. Figure 9.5 shows a visual representation of the results obtained on unseen test data at position (-1;-3) - one of the extreme points in the grid.

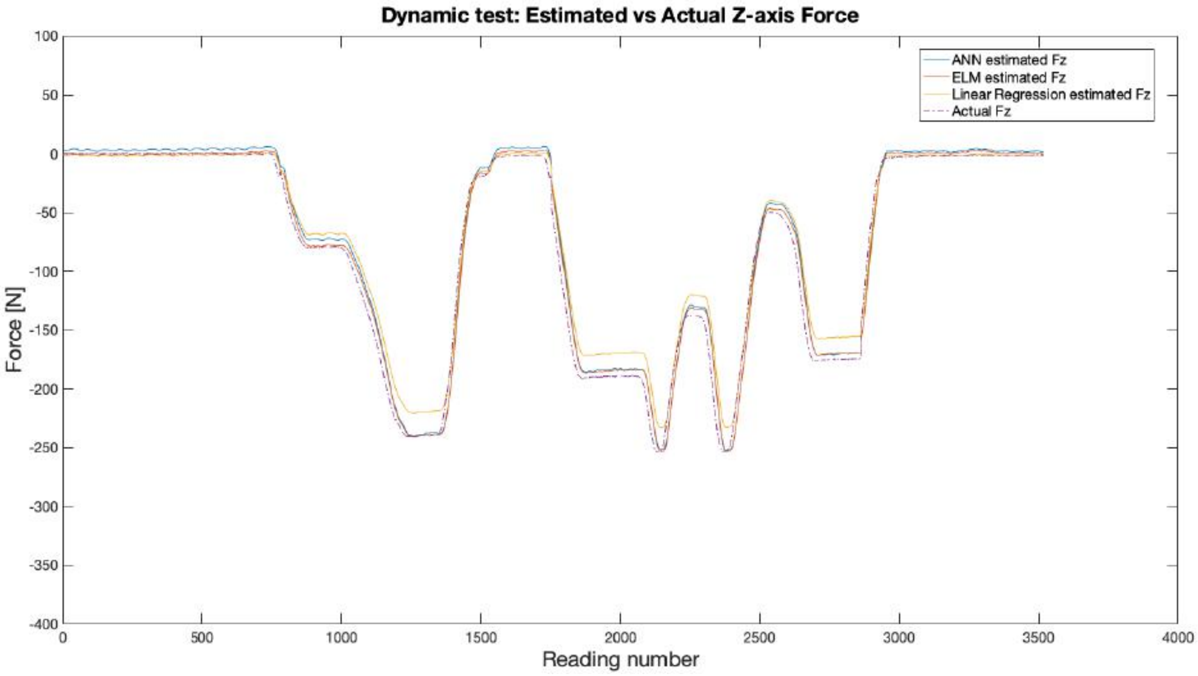


Figure 9.5: Test at extreme position (-1; -3)

Figure 9.6 below shows a heat map of the RMS errors from a neural network as a percentage of the full scale force (%FS) over the 26 testing points on the plate. Note that the $(x; y)$ co-ordinates shown in Figure 9.6 correspond directly to the already established plate co-ordinate scheme detailed in Figure 9.5. Heat maps showing data for ELM and linear regression errors are presented in figures A.2 and A.3 respectively.

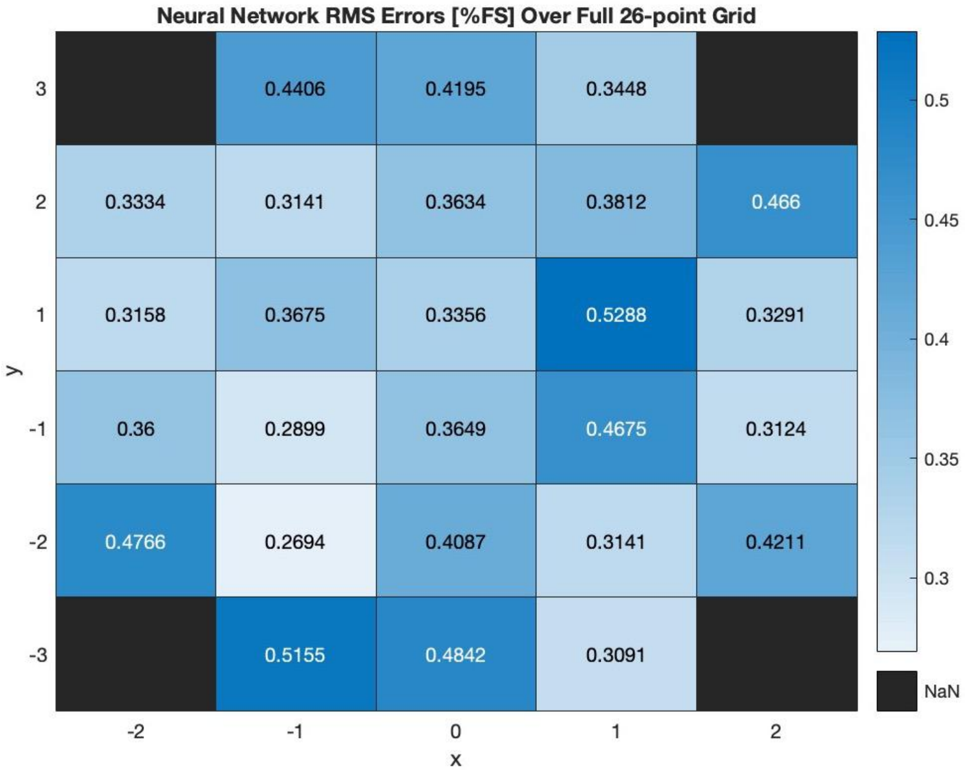


Figure 9.6: Heat map over the 26 switch positions showing the RMS errors of estimated *vertical* GRF by a neural network predictor. Errors are given as a percentage of the full scale vertical force of 2000N.

Positional interpolation

As described in sections 7.5.2 and 6.2.2, a discrete grid of 26 switches was used to capture xy position of the COP of the applied GRF. Figures 9.7 and 9.8 below show estimated responses to a random vertical force profile at two interpolated points: (-1.5; 0.5) and (-0.5; 0.5). These points are both examples of a position that required interpolation in both the x and y directions and were chosen in order to test interpolation near the edge and centre of the plate. Input interpolation entailed using the interpolated[4] $(x; y)$ co-ordinate as the input to the network. Output interpolation was achieved by presenting the same load cell data to the network four times but each time, using the four different known[5] $(x; y)$ co-ordinates of the grid points surrounding the interpolated point. The network outputs from these four known points were then averaged to give the output-interpolated result. Further interpolation testing results are presented in Section 9.4.1.

[4] i.e. a position that had not directly been exposed to the network during training

[5] co-ordinates that had been exposed to the network during training

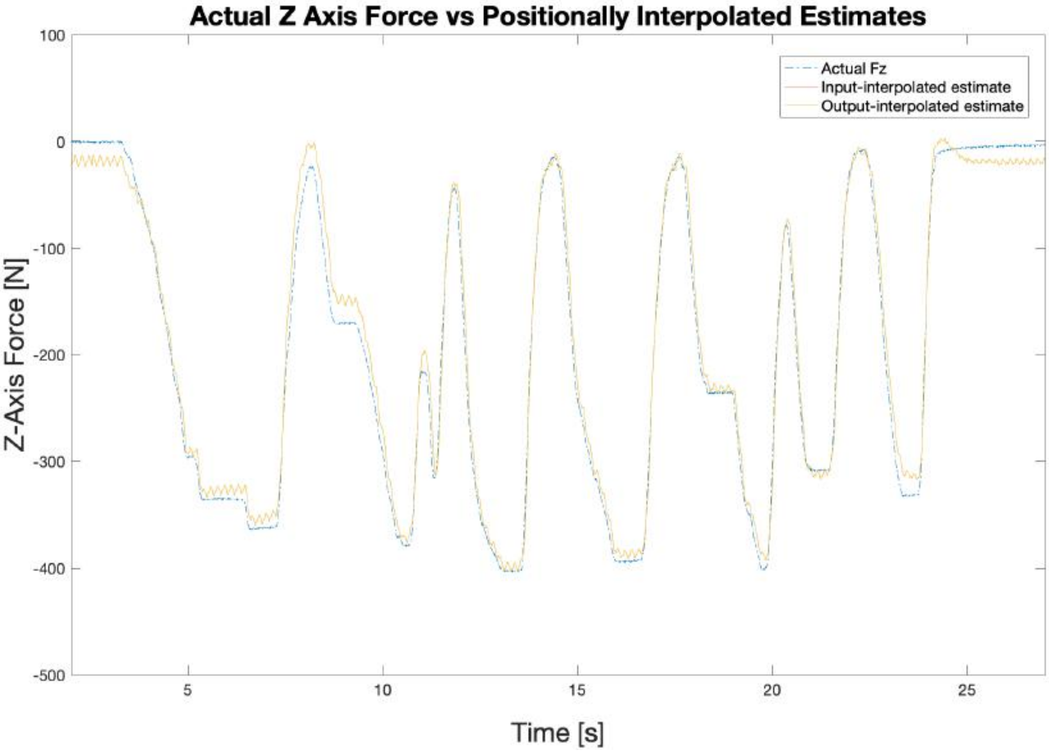


Figure 9.7: Graph showing network estimated vs actual vertical GRF Fz at interpolated position (-0.5; 0.5). Although hardly distinguishable, results from both interpolation strategies as discussed above are shown.

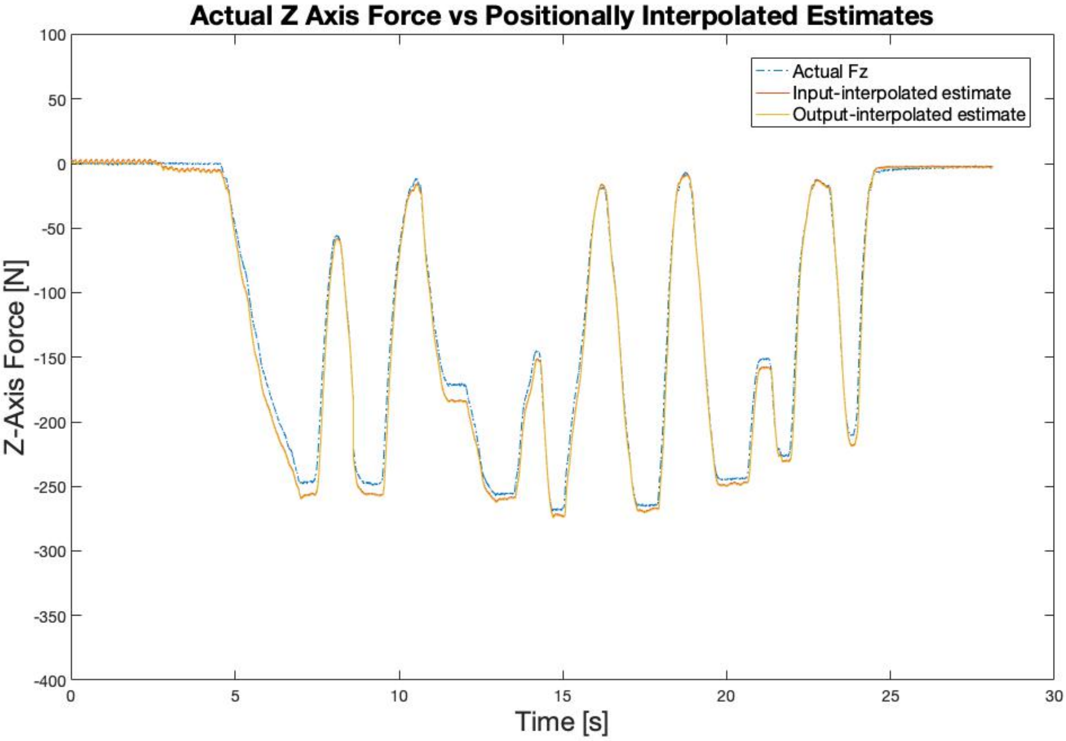
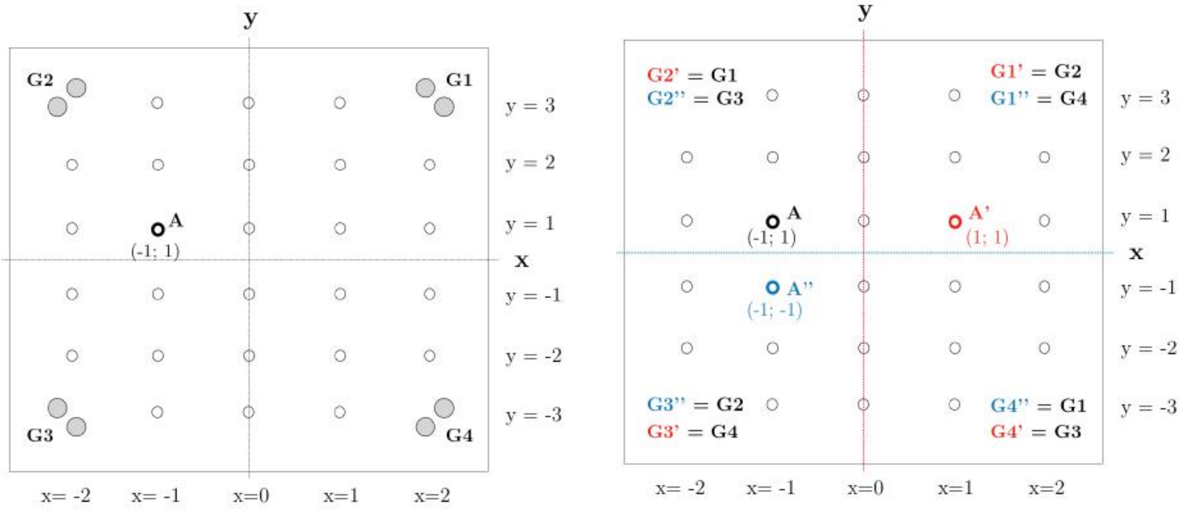


Figure 9.8: Graph showing network estimated vs actual vertical GRF Fz at interpolated position (-1.5; 0.5). Results from both interpolation strategies as discussed above are shown.

9.2.4 Geometric symmetry

Due to the symmetry of the plate about two sets of orthogonal axes (x-y axes and the $y = x$ and $y = -x$ lines), tests were conducted to determine if this symmetry could be exploited to generate artificial data points. This notion is clarified by the example in Figure 9.9 below.



(a) Initial test point A

(b) Points A' reflected about the y axis and A" reflected about the x axis

Figure 9.9: Diagrams showing the process of creating artificial test points using geometric symmetry of the plate. Notice the reflected points A' and A" and corresponding reflected load cell positions G_i' and G_i"

Figure 9.10 below shows the estimated force responses with unadjusted inputs (at the original point A in Figure 9.9).

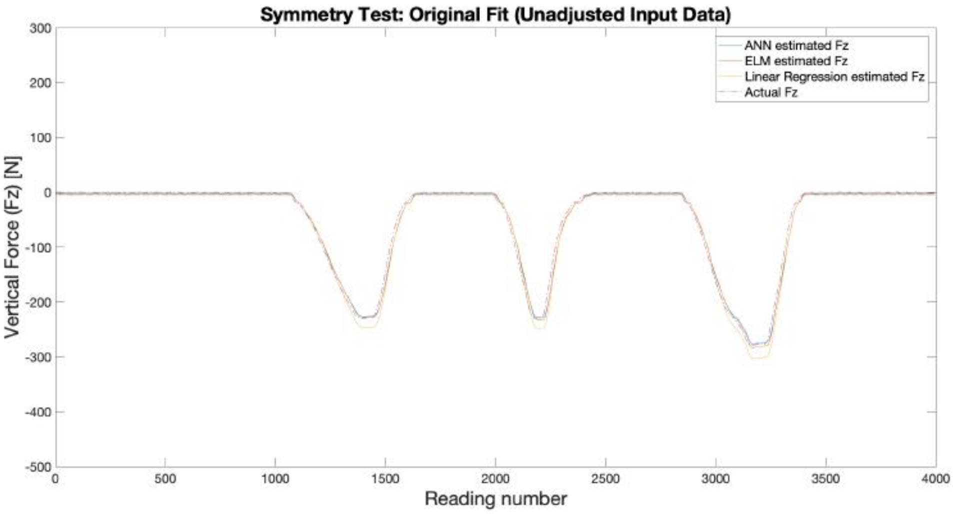


Figure 9.10: Graph showing estimated force responses with unadjusted input data at point A

Figures 9.11 and 9.12 below shows the estimated force responses with gauge positions rearranged as depicted in Figure 9.9 to simulate data collected at reflected points A' and A". Notice that the estimators are largely unable to closely predict the actual F_z force on artificially created points with gauges rearranged.

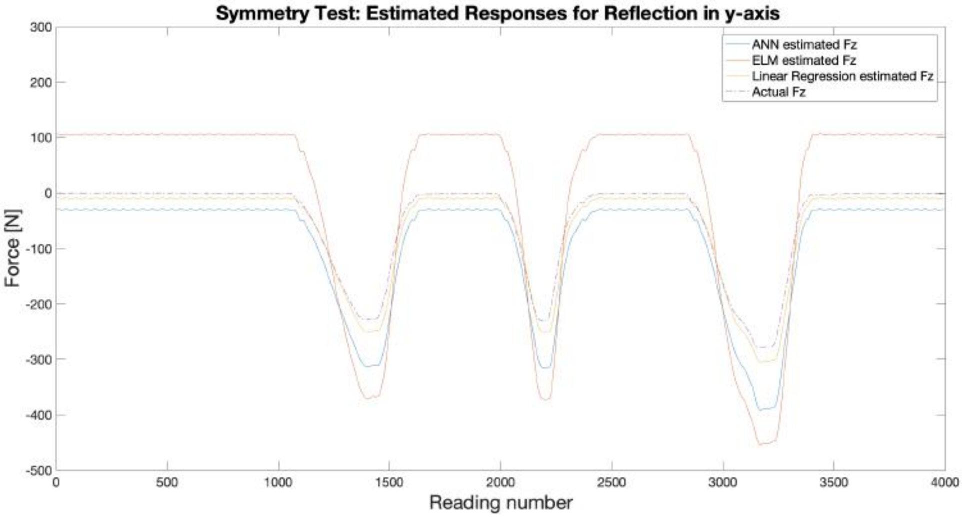


Figure 9.11: Graph showing estimated force responses at artificially created point A' (A reflected in y-axis)

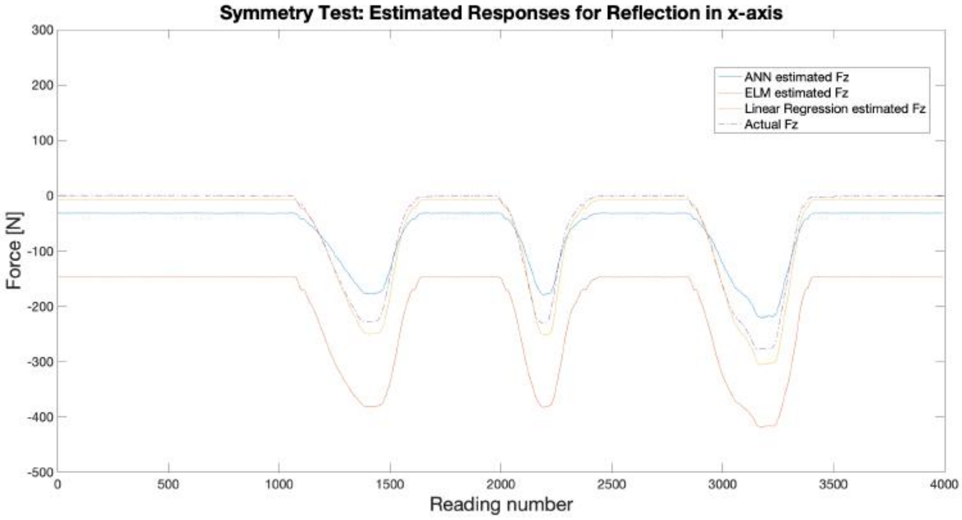


Figure 9.12: Graph showing estimated force responses at artificially created point A" (A reflected in x-axis)

9.3 Two Axis GRF Estimation

The following sub-section details the results of two axis (F_z and F_x) tests performed with the apparatus and procedure described in Section 4.2.2. It was decided that the same rigorous testing procedure performed in the single axis (F_z) tests would not be applied for testing 2-axis performance. Instead, further rigorous testing was first conducted for full 3-axis tests as this was deemed a more challenging task in general.

However, in order to maintain a systematic testing approach, the following experiments and their results are presented below. Figure 9.13 below shows the the estimated vertical and mediolateral (F_z and F_x respectively) forces applied near the centre at position (1;1) on the plate:

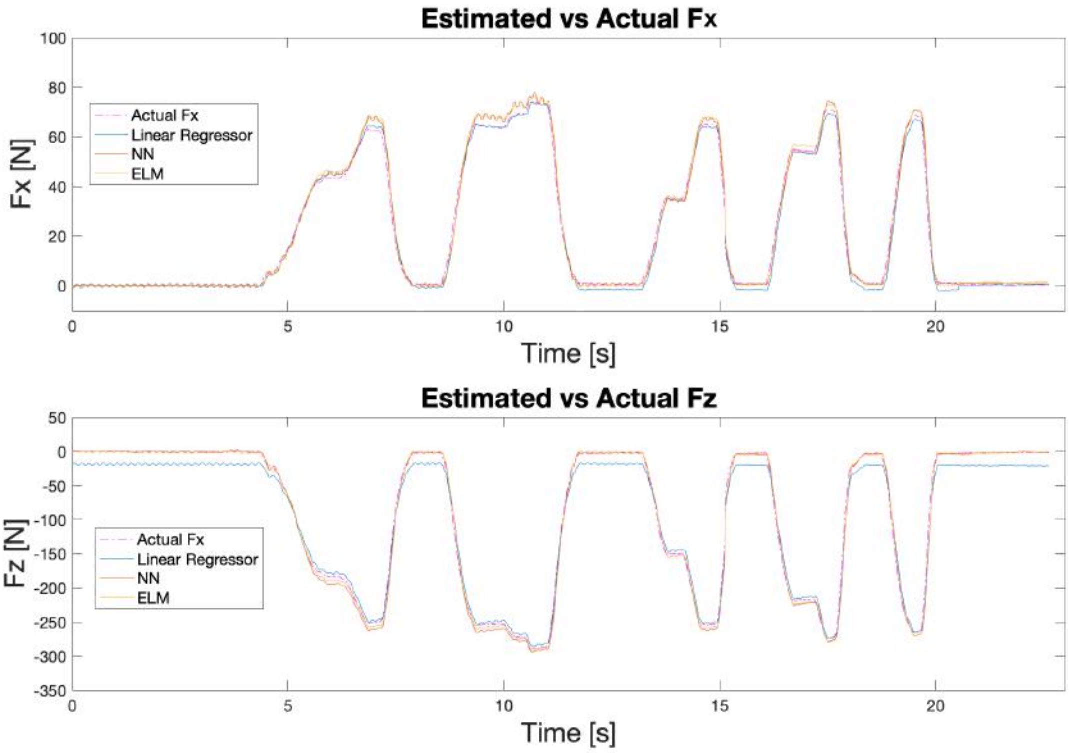


Figure 9.13: Graph showing estimated 2-axis force responses for an arbitrary dynamic load applied at position (1;1) on the plate.

9.3.1 Effect of Training Data on Predictor Performance

It was noticed that the nature of the training data presented to the various regression models had a significant effect of the prediction performance. Predicting the lateral x-axis force (F_x) always proved more challenging than the vertical component and was thus analysed independently for the following training tests. Figures 9.14 and 9.15 show the training data (and the force estimated using the training data) presented to a network. However, the two testing sets used on this network differ; Figure 9.14 shows the network's poor performance in estimating a large magnitude testing set and Figure 9.15 shows accurate prediction of a smaller magnitude testing set.

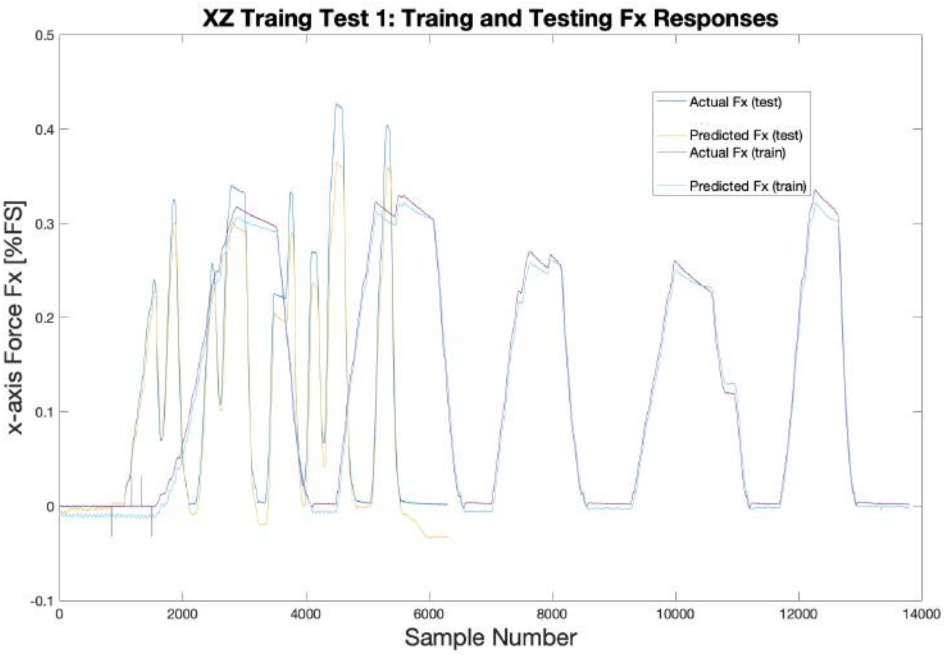


Figure 9.14: Graph showing poor F_x estimation for a testing set with values that exceeded several training set values in magnitude.

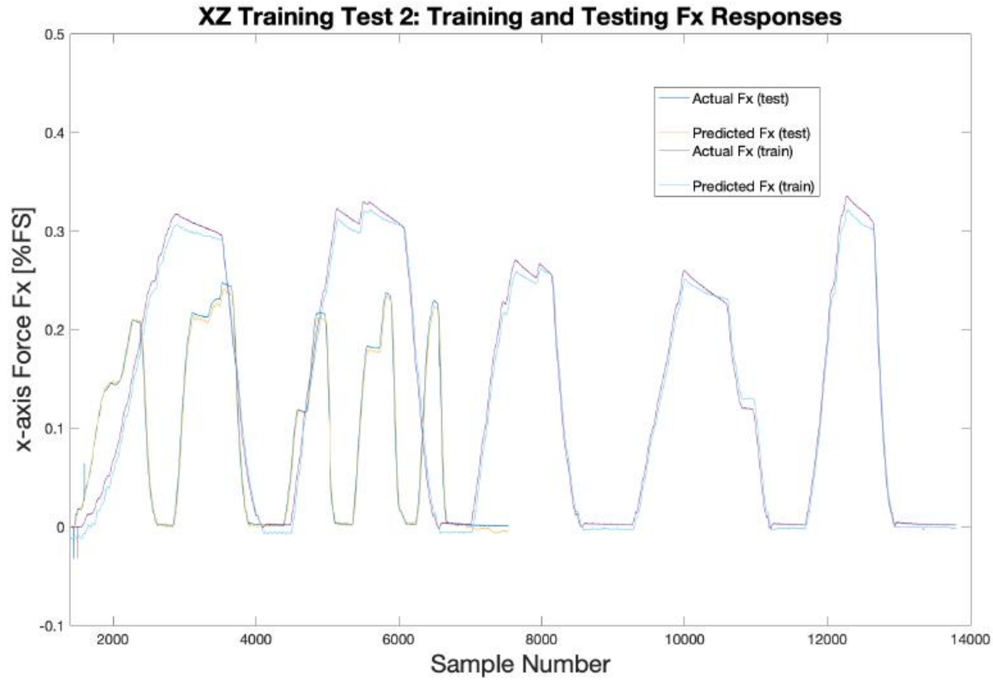


Figure 9.15: Graph showing accurate F_x estimation for a testing set with force magnitudes that were fully contained (smaller than) in the training set.

9.4 Three Axis GRF Estimation

The following sub-section details the results of full 3-axis tests performed with the apparatus and procedures described in Section 4.2.2, as well as some auxiliary tests. Figure 9.13 below shows the the estimated 3-axis GRF components for a multi-axial load applied near the plate centre at position (1;1) on the plate. Initial 3-axis data for both training and testing was collected using the testing rod with rubber foot as shown in Figure 4.5.

9.4.1 Positional Testing

As discussed in Section 9.2.3 and following similar structure to the test results presented in the single axis tests (Section 4.2.2) above, a 3-axis force dataset was acquired for each of the 26 switch positions on the plate. The same error metrics discussed in Section 4.2.2 are displayed for each of the three GRF components (Fx, Fy and Fz) in tables A.4 to A.5 in Appendix A. Heat maps corresponding to these results are shown for the NN model are also shown in A. Figure 9.16 below shows the estimated 3-axis GRF performance for position (0;1) - a point close to the origin (geometric centre) of the plate. As with each of these 26 positional tests, a linear regression model, ELM and neural network are compared by their prediction performance. Note the significantly improved performance of the ELM and NN models in comparison to the linear regression model.

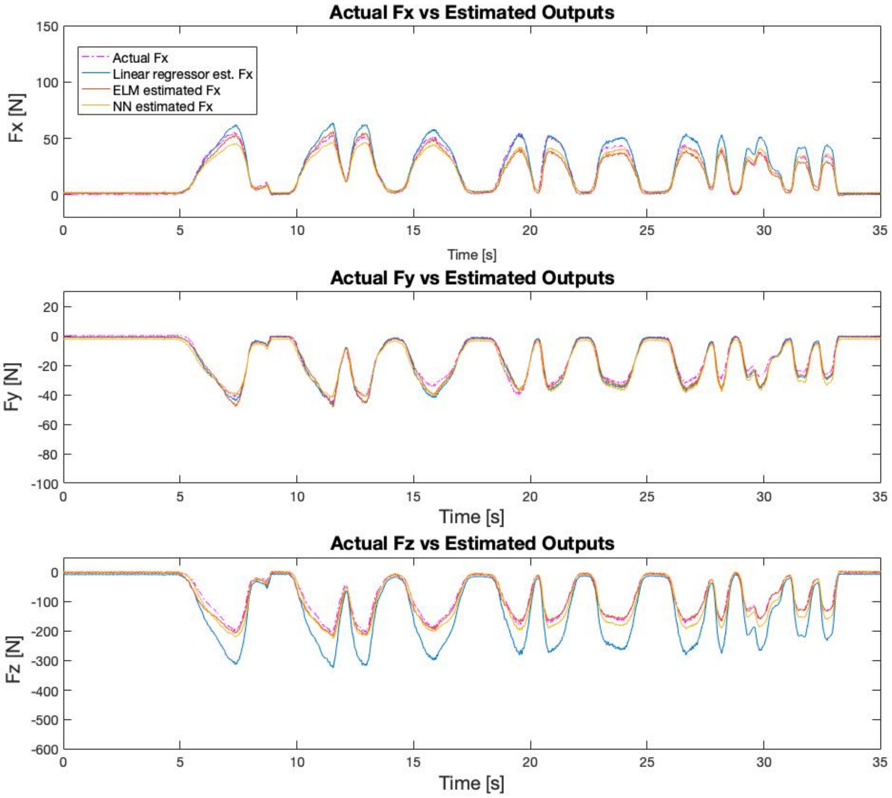


Figure 9.16: Graph showing estimated 3-axis GRF components against their true counterparts for an arbitrary dynamic load applied by the testing rod at position (0;1) on the plate.

Figure 9.16 below shows the results of the test which produced the poorest performance in estimating the 3-axis GRF components. This test was at position (3;-2) - an extreme point near the edge of the plate.

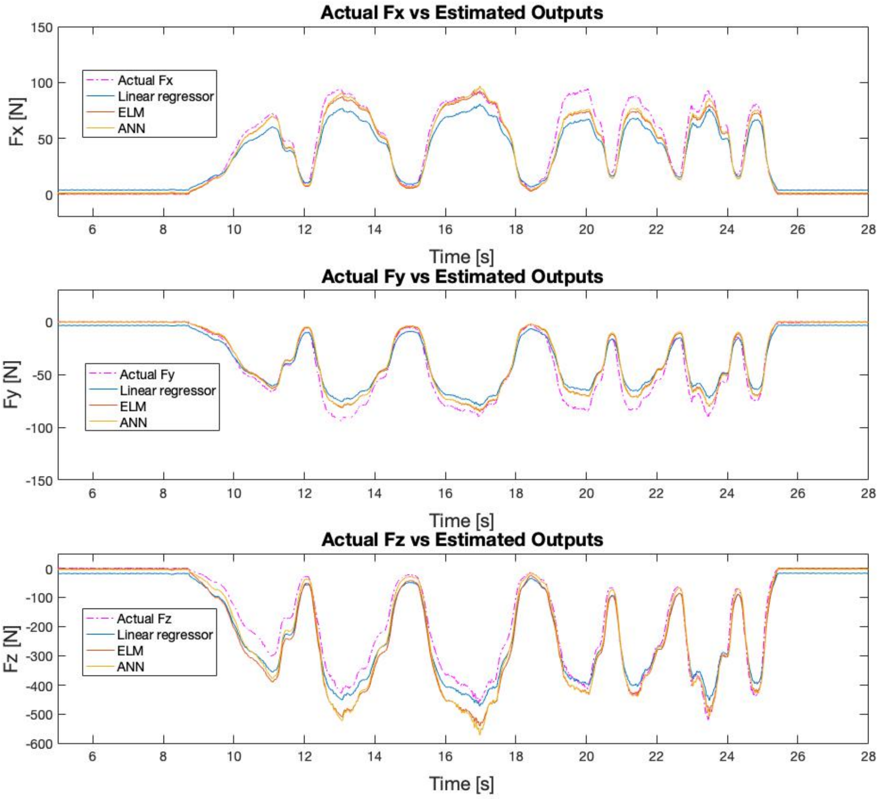
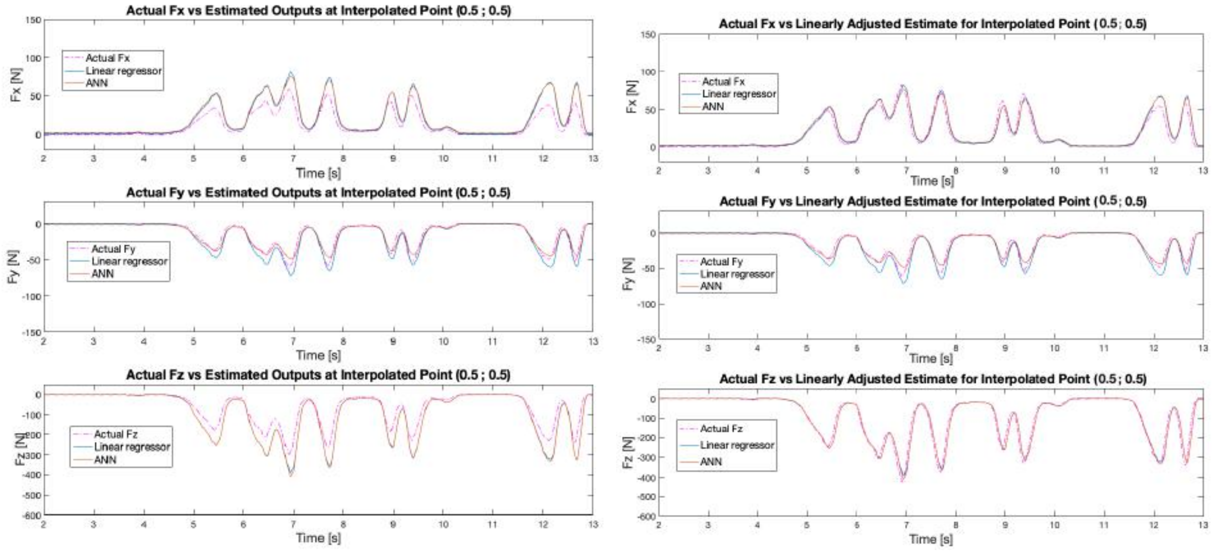


Figure 9.17: Graph showing estimated 3-axis GRF components against their true counterparts for an arbitrary dynamic load applied by the testing rod at extreme position (3;-2) on the plate. The legend seen in the top figure is consistent for F_y and F_z plots.

Positional interpolation

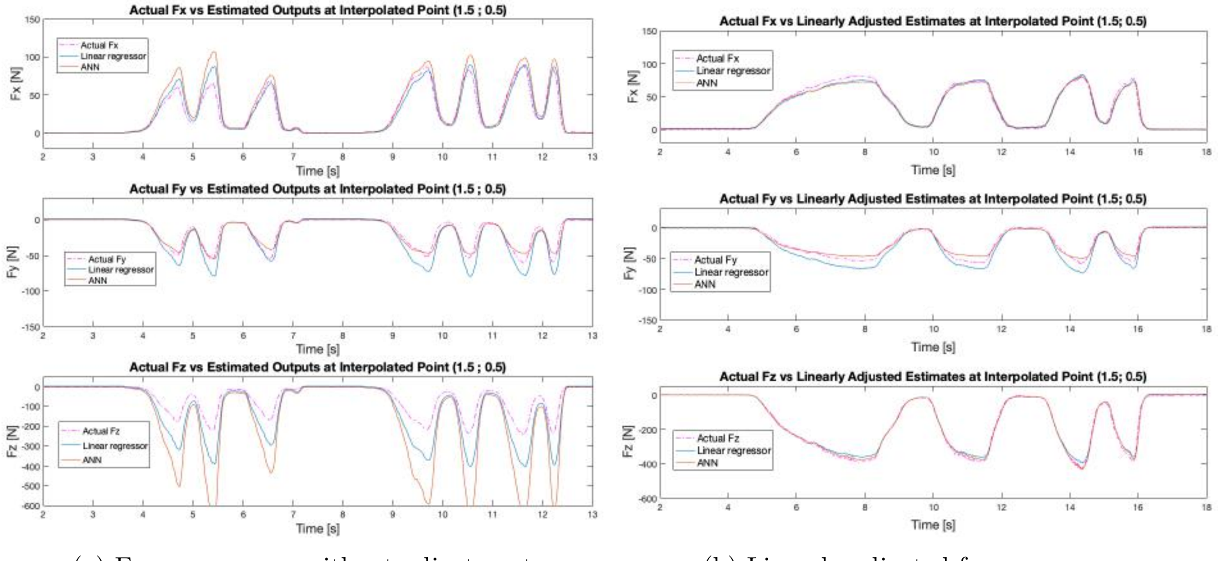
As described in Section 9.2.3 for single-axis interpolation testing, the platform's ability to estimate GRFs at interpolated points (in between the designated switch positions) was tested. For 3-axis versions of these tests, estimator performance at interpolated points was initially significantly poorer than that at designated switch positions (Figure 9.18(a) and 9.19(a)). However, it was noticed that the *underlying pattern* of the estimated GRF component force in each axis correlated very closely to its respective target force profile. By multiplying the estimated component force vectors by fixed scalar amount which was determined empirically, the improved estimation results results shown in 9.18(b) were obtained. Figure 9.19(a) and (b) follow the same structure but for a load applied at (1.5; 0.5). As with the single-axis experiments, input and output interpolation strategies yielded almost identical results and thus, due to its simpler implementation, the input interpolation strategy was applied here (i.e. the interpolated (x, y) co-ordinates were used directly as inputs to the regression models). Note that these interpolated *positions* did not form part of the training data and the models were never exposed to any data at these in-between positions.



(a) Force responses without adjustment

(b) Points A' reflected about the y axis and A" reflected about the x axis

Figure 9.18: Graphs showing the estimated GRF performance at interpolated point (0.5; 0.5) before and after linear adjustment



(a) Force responses without adjustment

(b) Linearly adjusted force responses

Figure 9.19: Graphs showing the estimated GRF performance at interpolated point (1.5; 0.5) before and after linear adjustment

9.4.2 Magnitude Testing

In order to ascertain whether larger forces (in the order of magnitude experienced by cheetahs mentioned in [22]) could still be estimated as accurately as in previous tests, larger magnitude training/testing data was first acquired. Typical results of a large-scale loading test performed are presented in Figure 9.20 below.

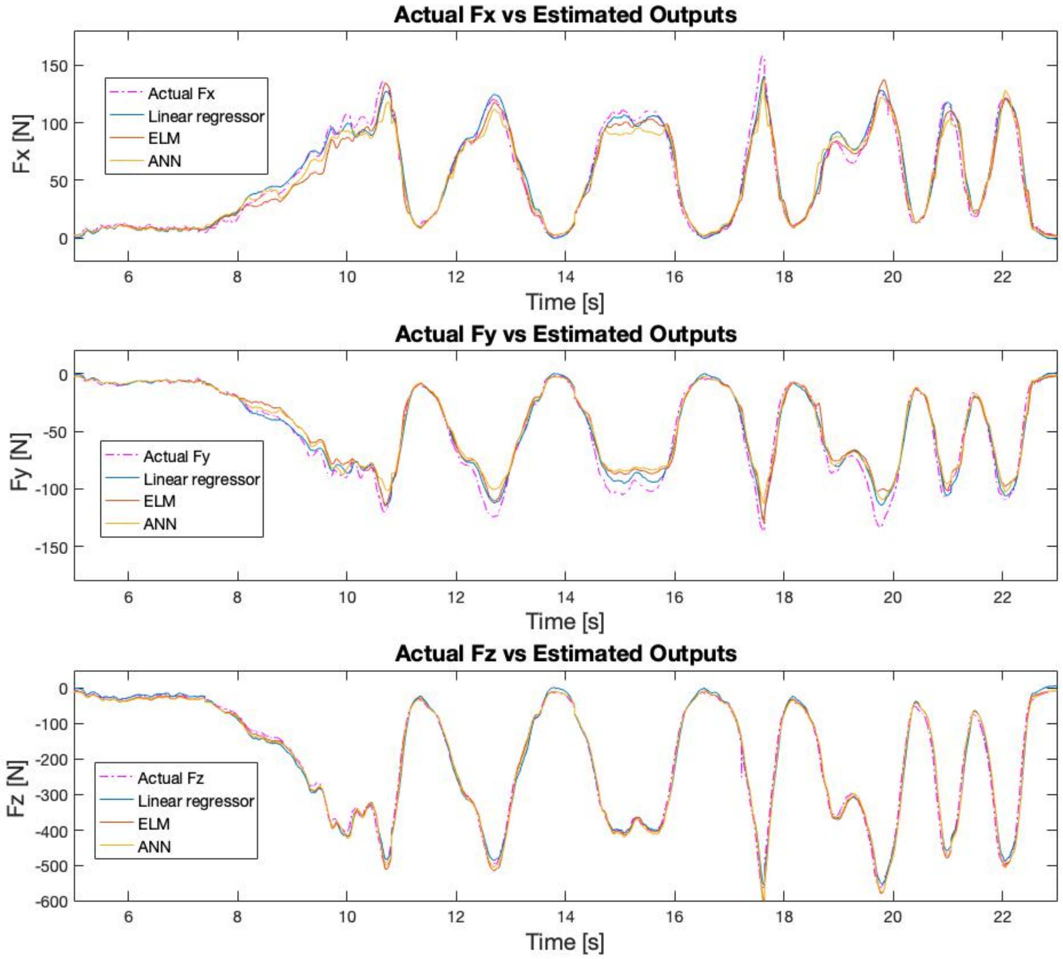


Figure 9.20: Graph showing estimated 3-axis GRF components for large-scale magnitude testing

9.4.3 Natural Gait Experiments

An important experiment was the plate's ability to estimate GRFs associated with natural human gait. As described in Section 4.2.2, a subject weighing approximately 50kg was requested to first step on the plate during natural walking and thereafter, during a short run.

Walking tests

Figure 9.21 shows the estimated GRF components of a neural network on data obtained during several steps on the plate during walking.

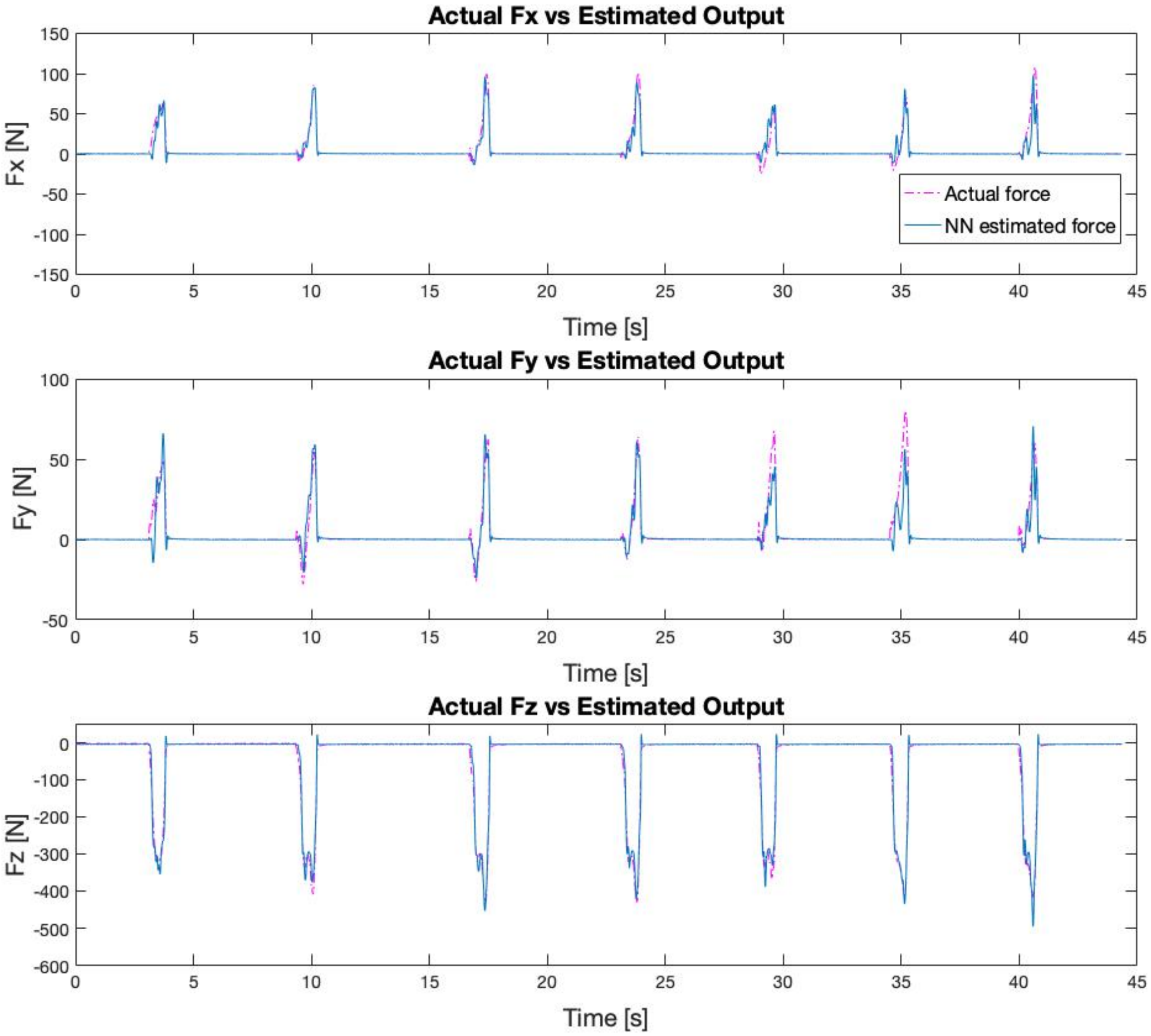


Figure 9.21: Graph showing network-estimated 3-axis GRF components and their actual counterparts obtained during natural walking

Table 9.2: Table quantifying the neural network prediction errors for data shown in Figure 9.21

Error Metric [%FS]

	RMSE	**SD**	**Max Error**
Fx	1.7141	4.1663	17.883
Fy	1.5447	3.6711	13.352
Fz	0.57534	1.1589	4.2733

Note that the standard deviation (SD) and max error metrics in Table 9.2 were computed on absolute errors (so as to prevent cancellation of positive and negative errors).

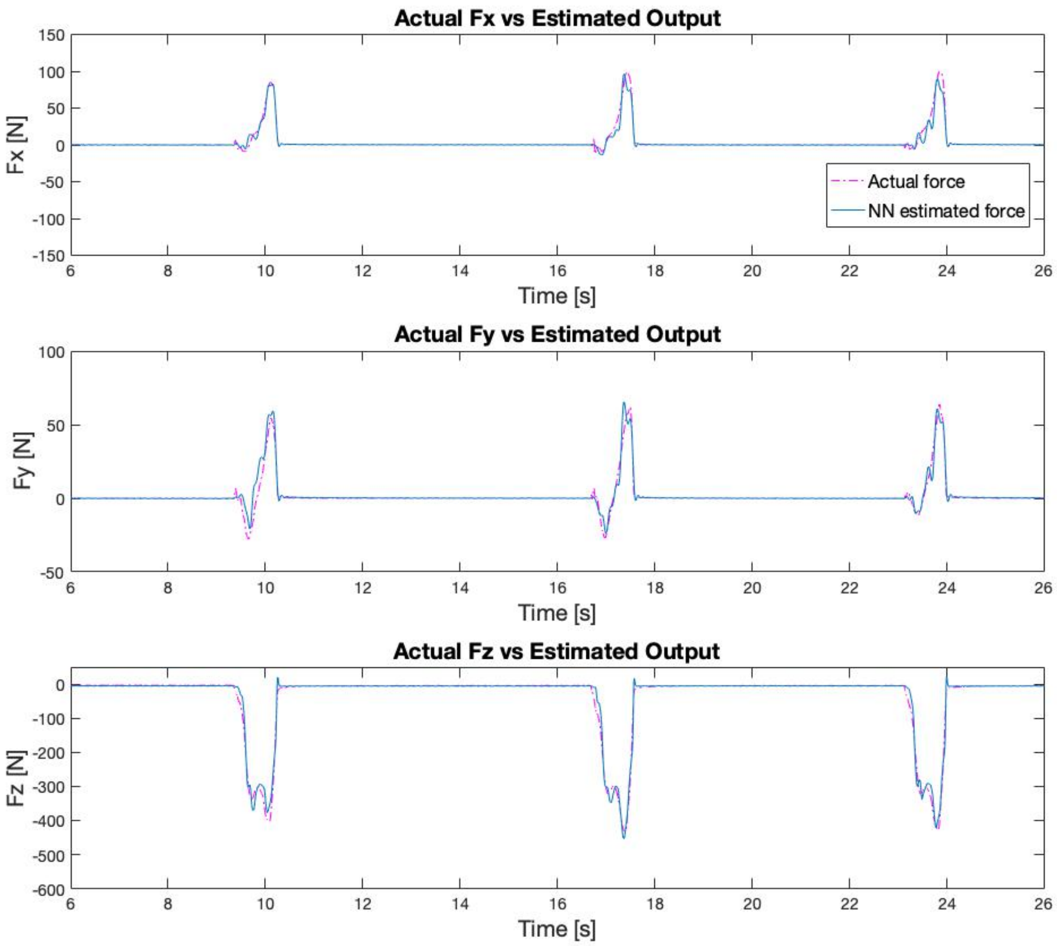


Figure 9.22: A closer look at the force profiles depicted in Figure 9.21 above

Note the similarity in the F_z and F_y GRF components (Figure 9.22 above) measured experimentally and the human gait force profiles recorded in the Parkinson's Disease study conducted in [9] (repeated in Figure 9.23 below for the reader's convenience). Note that compressive vertical forces were defined as *negative* in this project (opposite to the convention in [9] and thus Figure 9.23).

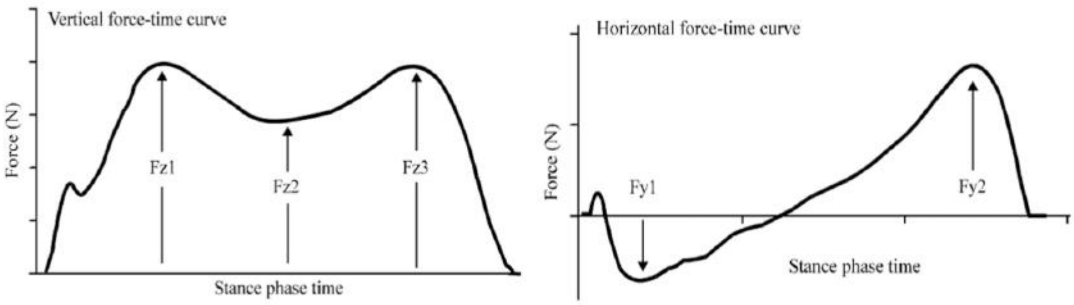


Figure 9.23: Graphs from [9] to show typical force-time curves for vertical (F_z) and anterior-posterior (F_y) GRF components for experiments conducted in this study.

The profiles depicted in Figure 9.24 show further similarity to the experimental results shown in Figure 9.22. Note also the presence of the mediolateral component in Figure 9.24 (which was not recorded in Figure 9.23) which also shows a similar profile to that in the result of this experiment.

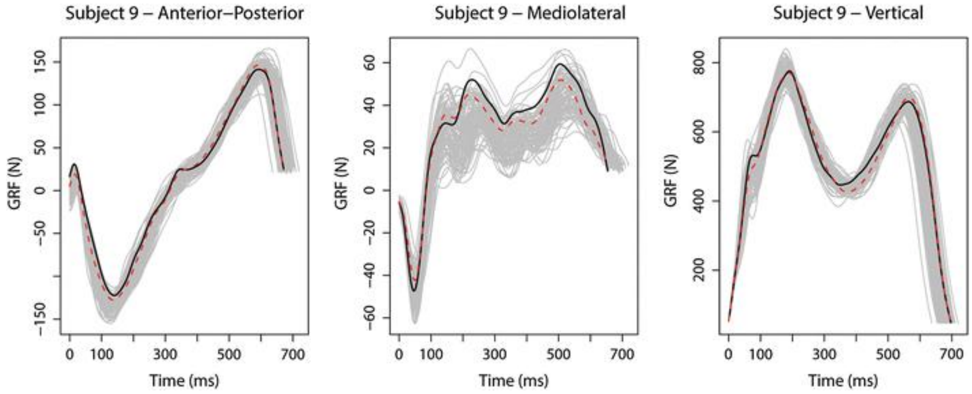


Figure 9.24: Graphs from [11] to show typical GRF profiles of human gait measured by novel nanocomposite piezo-responsive foam (NCPF) sensors. Results from several tests are shown by the grey curves and for a randomly selected test, the true GRF profile is represented by the solid blank lines and the estimated profile by the dotted red lines.

Figure 9.25 shows the predictor performance of an ELM and linear regressor for the same task and input data as in Figure 9.21 above.

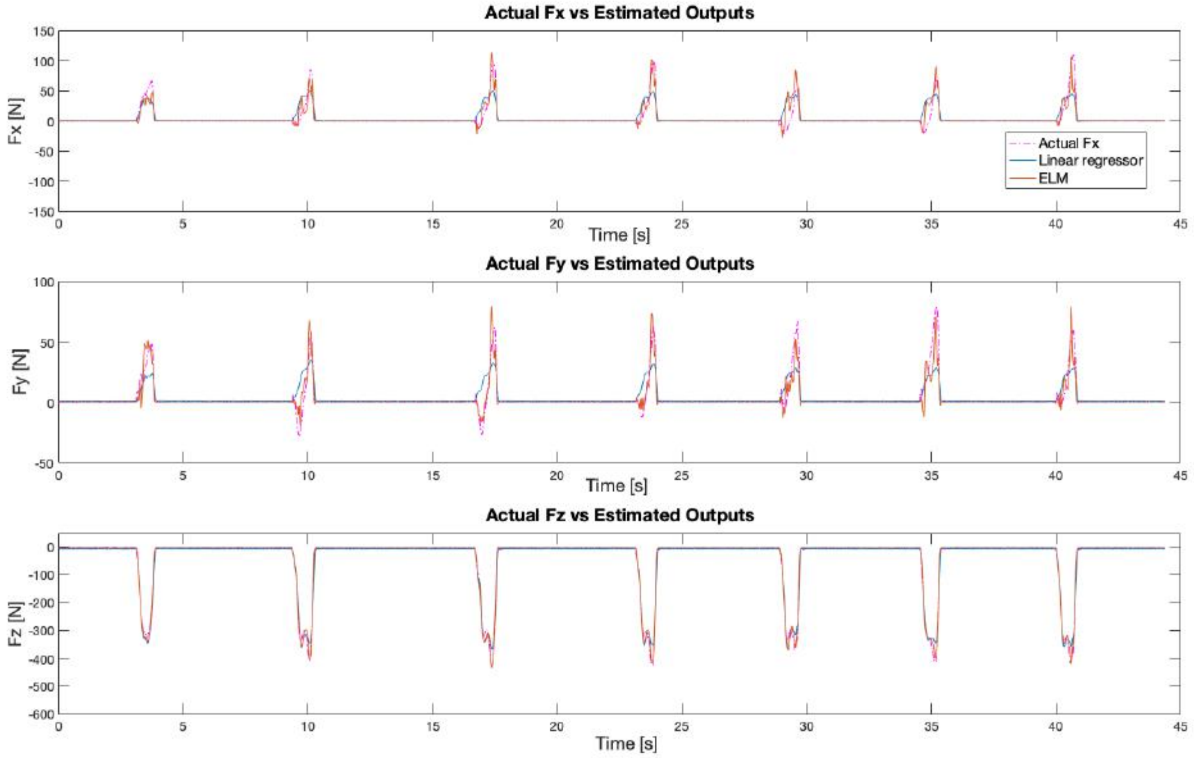


Figure 9.25: Graph showing 3-axis GRF components estimated by a linear regressor and ELM obtained during natural walking.

Table 9.3: Tables quantifying the ELM and linear regressor prediction errors for the data shown in Figure 9.25 above

Table 9.4: ELM prediction errors

	Error Metric [%FS]		
	RMSE	**SD**	**Max Error**
Fx	2.2868	5.2635	20.386
Fy	1.6504	3.7422	16.386
Fz	0.33523	0.7154	3.0411

Table 9.5: Linear regressor prediction errors

	Error Metric [%FS]		
	RMSE	**SD**	**Max Error**
Fx	3.1819	6.5034	22.329
Fy	2.4446	5.0787	17.047
Fz	0.60488	1.2306	3.9259

Running tests

After data for natural walking patterns on the plate was collected, the same 50kg subject as in the previous section was requested to step on the plate during a short run. The performance of estimated GRFs by a neural network is depicted in Figure 9.26. Figure 9.27 shows a closer look at GRF components estimated by a linear regression and an ELM model on a different data set acquired during tests with the same subject.

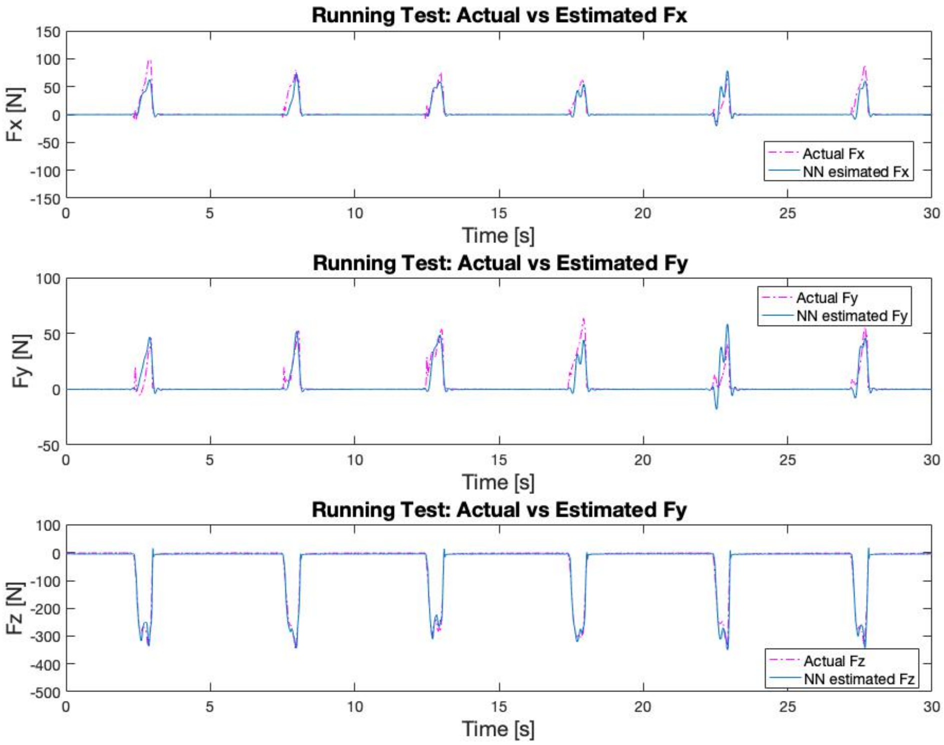


Figure 9.26: Graph showing 3-axis GRF components estimated by a neural network for impulsive forces applied through stepping on the plate during a light run

	Error Metric [%FS]		
	RMSE	**SD**	**Max Error**
Fx	2.123	4.508	13.641
Fy	1.517	3.166	10.038
Fz	0.36	0.695	2.304

Table 9.6: Table quantifying the neural network prediction errors for data shown in the running experiment in Figure 9.26

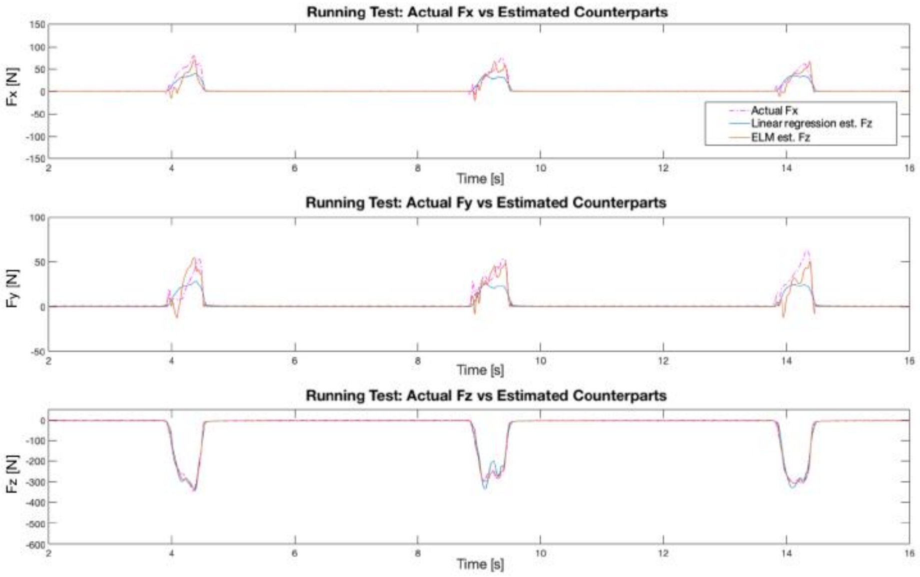


Figure 9.27: Graph showing a more detailed view of 3-axis GRF components estimated by a linear regression and ELM model for impulsive forces applied through stepping on the plate during a light run

Chapter 10

Discussion

10.1 Performance of the Data Acquisition (DAQ) System

A critical aspect of this system was the data acquisition system and the integrity of the data acquired from it - particularly the data from the four AD7730 devices coupled to the load cells. Without consistent and reliable data, even the most sophisticate machine learning models would have been rendered useless. Figure 9.1 provides an indication of the quality of the data received from the AD7730 devices. The extremely low level of noise in zero readings presented in this figure was observed to be consistent across the entire operating range of the load cells. This likely allowed *less data* and *simpler* machine learning models to perform well due to the fact that data presented to the models did not require filtering through learning of extensive volumes of data for the same input behaviour.

Unfortunately, the PCBs designed in Section 6.2.3 were not manufactured and returned for experimentation before the end of this project (despite the design and fabrication files being submitted to the manufacturer, TraX, more than a month prior to the end date of this project). Consequently, the performance of the DAQ system using the PCBs in place of the prototype board depicted in Figure 6.7 could not be analysed.

10.1.1 Verification of Acquired Data

Although the DAQ system seemed to perform extremely well from initial samples, it needed to be ascertained whether the sampled data was *meaningful* or not. From Figure 9.2 which plots the raw load cell readings against the actual vertical force measured by the OptoForce (OF) sensor, it was seen that the unprocessed load cell data acquired from the DAQ system correlated extremely closely to the 'truth' data from the OF sensor. Subtle patterns in the measured force were captured by the DAQ system over large ranges of the input space and furthermore, the force profiles show excellent synchronisation in time (even though the load cell readings and OF data were sampled and retrieved from two independent sensor systems).

10.1.2 Precision and Consistency

The results of the static loading test depicted in Figure 9.3 provide an indication of the precision and consistency of the DAQ system. A vertical force induced by a fixed 6kg mass was accurately estimated in successive placements of the mass on the plate. Small spikes and oscillations in the data acquired from the OF sensor can be attributed to the fact that the mass was placed somewhat inconsistently by hand during experiments. Furthermore, the small vertical force induced by a 6kg mass could be estimated extremely accurately even though this system was designed for loads in the region of 30kg - 100kg (for applications with humans and animals such as the cheetah).

10.2 Performance of Regression Models

In this section, the results presented in Chapter 9 characterising the performance of the various regression models are analysed. Model performance was most commonly numerically characterised by observing the RMS error (RMSE), standard deviation of the absolute error and maximum absolute error. It should be noted that RMSE formed the most commonly observed metric in the literature [23, 24, 26, 41, 40] for evaluating regression model performance and greatest attention should be paid to it. The standard deviation of absolute error is provided to give an indication of the spread of the error about the (absolute) mean value. While the maximum error is also indicated in each error set, this metric should be considered with caution as even one outlier or erroneous sample (typically out of several thousand) in the estimated (or target) force profiles would skew the error figures with a disproportionately large maximum error.

10.2.1 Static vs Dynamic Performance

As shown with single-axis (vertical) loading results in Section 9.2, the various regression models were able to estimate dynamic[1] forces equally well as - and often better than - static loads. It is believed that this phenomenon is due to the fact that machine learning models were provided with no indication of the temporal link between samples. This is because, when presented a data set of x discrete samples, the samples were randomly divided and shuffled into training, testing and validation sets. For every sample, each model would simply try use the instantaneous load cell and position inputs to match[2] them to the target force component outputs *at that instant.* Hence, whether successive readings (of inputs and corresponding outputs) were constant or rapidly changing was of no consequence to the models. In fact, static or gradually-changing

[1] characterised by a constantly changing force gradient

[2] by evaluating the cost function to assess prediction performance and consequently adjusting relevant internal model parameters

force profiles were often estimated *less successfully* than their dynamic counterparts since in static profiles, a particular value (or values within a very small subset of the domain) would often need to be estimated over many successive samples. If the models did not receive sufficient exposure to this particular value (or small subset) during training, they would likely fail to estimate the corresponding outputs as accurately and would accumulate significant offset error over many readings as a result. Due to their rapidly changing behaviour, this issue was less prevalent with dynamic loads. This is encouraging as force profiles in almost all foreseeable applications of this platform would likely be dynamic and impulsive in nature (as documented in [2, 3, 5]).

10.2.2 Analysis of Positional Tests

An important practical requirement of this system is that it should be able to estimate forces applied at any point on the plate. As shown in the results of single axis and three axis positional tests in Section 9.2.3 and 9.4.1 respectively, the plate was able to estimate GRFs remarkably well at each of the 26 switch positions across the plate.

Of course, tests with data recorded on certain positions yielded better results than others but interestingly, there was no clear relationship between the prediction error and the distance of the COP of the applied load from the centre of the plate. Heat maps such as those presented in Figure 9.6 and Section A.3 showed a positional distribution of errors that is almost random - RMS prediction errors are not minimal at the centre of the plate and the errors do not increase predictably towards the edge of the plate. This may be due to the fact that there were slight differences in magnitudes of load cell readings, even when a vertical load was applied at the centre of the plate. The random error distribution suggests that the decreased errors at certain points were likely a consequence of a more representative or effective training data set recorded at that position. This result is potentially encouraging as it does not suggest that the performance of this design is limited to the centre of the plate.

10.2.3 Force Magnitude and Model Performance

There was no indication from the experiments performed that an increase in the applied force on the plate led to poorer or better model performance in estimating GRFs. Figure 9.20 shows that the system was able to accurately estimate forces up to approximately 150N in the lateral axes and 600N in the vertical axis. With the exception of the vertical component, these forces are in line with the maximum expected GRFs experienced by the cheetah in the study documented in [22]. Furthermore, the only reason larger forces were not tested was that the available testing apparatus made it difficult to acquire a sufficient set of reliable 3-axis readings of larger magnitudes. Given the evidence of consistent performance from experiments over a wide range of forces, it is believed that the maximum forces that can be estimated by the system are only mechanically limited by the sensing elements used (in particular, the load cells) and not the machine learning or electronics systems.

10.2.4 Training Data Strategies

The results of the two-axis training data experiments documented in Section 9.3 provide insight into a particularly important concept discovered during experimentation. When a model was trained using training data with a maximum magnitude of F_{max}, it was largely unable to estimate forces that exceeded F_{max} in unseen testing data (as seen in Figure 9.14). However, as shown in Figure 9.15, a model trained on the same data set was able to estimate unseen forces that did not exceed F_{max} almost perfectly. This is likely due to the fact that the system dynamics were non-linear across the full input-output range (domain) and even if a model was given an opportunity to learn the system dynamics over one part of the force domain (such as for training data containing forces between 0 and 30% of the full scale range), it would not necessarily be able to extrapolate to other parts of the domain (such as the 40% to 50% range).

It was also discovered that training the models on gradual, non-impulsive data (like in the force profile depicted in Figure 4.6) yielded models capable of significantly better prediction performance on unseen data. The reason for this fairly obvious; given more instances of a particular training point (with associated input and output force data), a model will better learn the input-output mapping for that point (and likely for dynamics within the neighbourhood of that point). It was deduced that ideally, training data should expose the models to a near-uniform distribution of the expected input-output loading range.

10.2.5 Positional Interpolation Capability

As discussed in Chapter 9, experiments were conducted to determine whether the platform could accurately estimate GRFs applied at 'interpolated' points in between designated switch positions. This is an important capability as a human foot (or possibly even a cheetah foot) would likely depress/trigger several switches when stepping on the plate. Although, in such a case, the resulting COP of the distributed load may well coincide with a designated switch position, it certainly would not always be the case. As a result, the worst case position[3] in the centre of a square of switches was tested near the edge and centre of the plate. Figures 9.7 and 9.8 show that for purely vertical loads, GRFs at interpolated positions could be estimated with a degree of accuracy comparable to the those applied at designated switch positions (i.e. positions which were exposed *directly* to models during training).

For 3-axis interpolation tests, Section 9.4.1 showed initially poor estimator performance at interpolated points but outlined how estimator outputs could be linearly adjusted to achieve comparable performance as at designated switch positions. Importantly, these linear adjustments (offsets) were consistent for several loading tests at a given interpolated position and thus, given

[3] the position furthest from a designated switch position which would require the largest interpolation from the regression models

testing data at several interpolated positions, the corresponding linear adjustments could be determined empirically and stored in look-up table.

This, however, is not the ideal solution and several other options of overcoming this problem exist. It was noticed that if training data was recorded for forces applied at an interpolated point and then used in model training, the models would be able to estimate new forces (in unseen test data) at that interpolated point equally as well as with forces applied at designated switch positions. Clearly, the implication is that *training* data should be taken at more positions on the plate. Although, ideally, a continuous xy sensing mechanism should be implemented to determine the COP of the applied GRF, even the existing discrete 26-switch grid would be more effective with training data recorded with finer positional resolution.

10.2.6 Analysis of Symmetry Testing

The results of tests conducted in Section 9.2.4 showed that synthesising an 'artificial data set' by reflecting the data point co-ordinates and appropriately rearranging the order of the load cell inputs presented to the models (as shown in Figure 9.9) was ineffective. This is likely due to differences in the load cells and particularly, in the way they were mounted - the load cells were secured to their respective mounts with nuts and bolts that were inconsistently tightened. Furthermore, although the PGA in each of the AD7730 chips was set to the same gain, the relative magnitudes of readings between the four load cells was never quite consistent - even for a theoretically even vertical force applied at the centre of the plate. It is unclear whether this was due to a slight issue with the data acquisition system or the actual load cell sensing elements.

10.2.7 Analysis of Natural Gait Results

In Section 9.4.3, it was shown that the plate was able to accurately estimate all three GRF components experienced during natural gait of a human subject. The predicted and actual force profiles depicted in figures 9.21 and 9.22 show an exceptionally close match. Furthermore, these results show prediction errors in line with the three axis positional tests conducted with significantly more gradual forces and more data. This shows that the system is able to competently estimate *impulsive*, sharp forces (to be expected in the study of animal motion and human sports science applications alike). It also supports the theory that the rate of change of successive data samples makes no difference to the regression models as even with significantly less training data for walking tests, the models were able to accurately estimate highly impulsive forces with rapid (large) changes between successive samples.

The specific profile of the the vertical (F_z) and anterior-posterior (F_y) GRF components shown in Figure 9.22 is particularly worth noting. Figure 9.23 below it shows the corresponding human gait F_z and F_y force profiles recorded during experiments conducted in [9]. These curves show an exceptionally close resemblance to the corresponding profiles recorded by the system in this project when subjects were requested to walk on the plate. Even the more pronounced take-off spike (labelled F_{y2} in Figure 9.23) compared to the heel strike (F_{y1}) seen in Figure 9.23 is consistent with the F_y curve shown in Figure 9.22 in this project. This correlation between experimental results and those documented in the literature was observed again in Figure 9.24 extracted from [11]. This profile obtained from a novel piezo-based sensor further validates the observations above and shows similarity to the experimental result in the mediolateral GRF component.

The mediolateral (F_x) components present in the force profile in Figure 9.22 can be explained by the fact that the subject was requested to walk primarily in the direction of the plate's y-axis but was instructed to allow for lateral motion (in the x direction) when pushing off the plate to introduce mediolateral F_x components into the data. This was done to test the system under the most challenging scenario where all three GRF components were present as this would likely be the case in practical applications with an animal or human without the constraint of having to walk or run along a single lateral axis. The F_x components seen in Figure 9.22 increase sharply towards the end of the stride and are consistent with the instruction to allow lateral (x-axis) deviation in the *push-off* motion.

10.2.8 Choice of Machine Learning Models

The default neural network (NN) as discussed in Chapter 8 was most commonly used as this architecture was determined to be the most robust and performed best across different tests. This decision was guided by the MIT study in [26] where a neural network with 10 hidden neurons proved to be most successful across various tests [26]. The effect of adding more hidden neurons was also investigated and as seen in Figure 10.1 below, networks with more than 10 hidden neurons did not necessarily perform better on unseen test data. In fact, the y-axis RMSE and max error (on unseen test data) can be seen increasing steadily with number of hidden neurons.

Furthermore, due to time constraints, it became unfeasible to train and retest several models with varying parameters due to the number of experiments performed. It was also decided that the performance achieved was more than adequate for a proof-of-concept model of this system and time was thus prioritised for testing more features and capabilities of the plate.

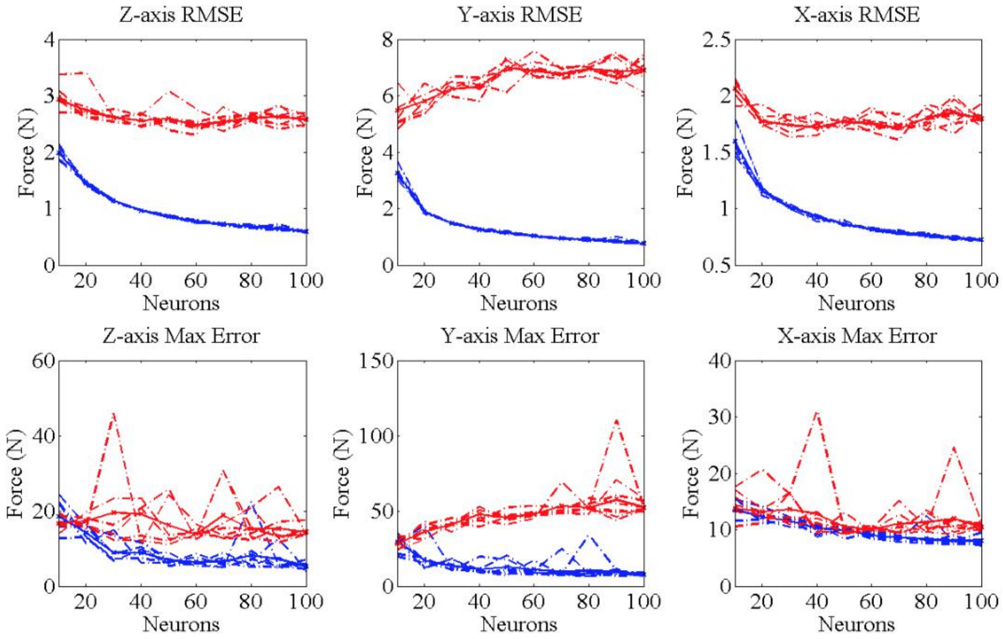


Figure 10.1: Figure from [26] depicting the effect of increasing hidden neurons in a neural network regression model. The blue curves show network performance on training data and the red curves show performance on testing data for 10 trials.

However, as alluded to in the theory development behind the ELM in Section 3.3, the ridge factor λ could be tuned very quickly to determine the optimal value for generalisation of ELM models across data sets. A numerically-optimised ridge factor that was most commonly chosen due to a good balance of generalisation and predictive power was $\lambda = 10e\text{-}3$. Comparing the prediction performances of various models presented in Chapter 9, it is evident that the ELM model developed from scratch performed comparably to the MATLAB neural network model and occasionally better too. Furthermore, the time taken to train ELM models was, in every case, several orders of magnitude less than a comparable neural network trained with the Levenberg-Marquardt backpropagation algorithm (which, itself, was considerably faster than the Bayesian Regularisation algorithm).

While the performance of the basic linear regression model was often significantly poorer than the NN and ELM models, it was included in each test as an *indicator of the complexity* of the task. In single-axis tests, the linear regressor was able to achieve almost identical performance to the other models but in the more challenging tasks, such as in the 3-axis gait experiments, it was notably inferior to the other models.

Although briefly discussed in Section 3.1.2, the SVM regression was not used extensively in this project. This is because the SVM regressor with a linear kernel gave almost identical results to the linear regression model in each test and the Gaussian kernel SVM regressor, although occasionally produced particularly impressive results, was very inconsistent and often produced oscillatory responses that would obscure the other results. Additionally, all SVM models took considerably longer than any other type of model to train (often up to 5 times as long).

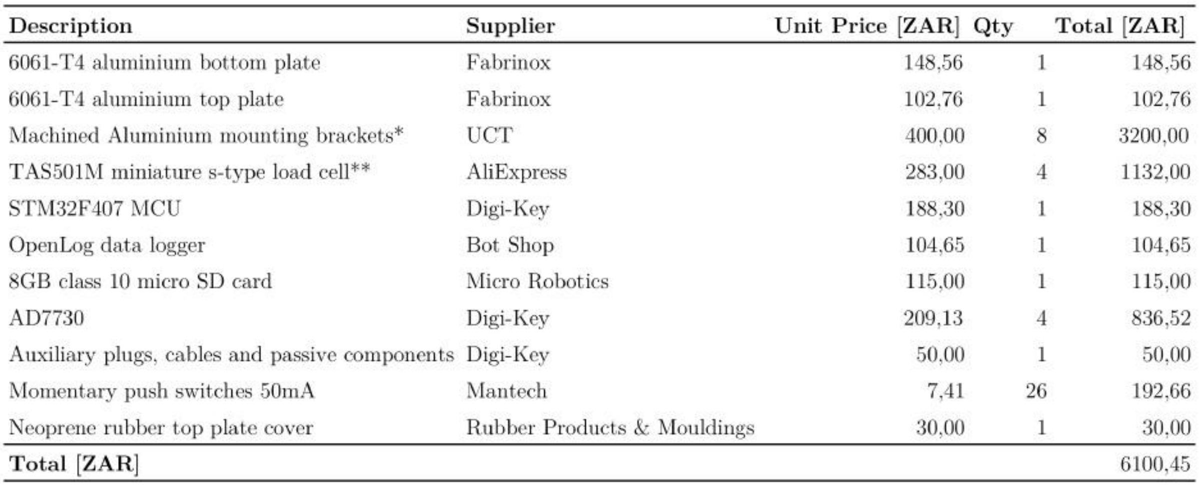
10.3 Limitations of the System and Experimental Procedures

A limitation of the discrete switch matrix used to determine the COP of the applied force would arise when trying to determine the COP of non-uniform pressure/force distributions at multiple points on the plate as might be the case during contact with a flat human foot. As the switches are only able to produce a digital signal with no indication of force, one can only assume that the COP of a distributed load lies at the centroid of the n switches pressed during contact with the body applying the force. However, the COP may not actually coincide with the geometric centroid of the active switch positions if the applied force distribution is non-uniform.

It should be noted that, referring to Figure 9.21, not all steps recorded showed a profile as clear as the plots in Figure 9.22. This can likely be attributed to the fact that, as discussed in Section 4.2.2, the OF sensor was required to be placed on top of the plate and the subject was requested to step on the OF sensor during their stride in order to gather target force data most accurately. Due to the limited surface area of the OF sensor, the subject was not always able to step on the sensor with a full roll of the foot during landing and take off.

Furthermore, the performance of this system is measured, in all cases, with the OF sensor as a reference which is assumed to be *accurate*. Although, practically, this sensor has its own internal error which is likely not negligible, the regression models are not influenced by this and they would perform equally well if the truth data was infinitely accurate (their task is just to track a known reference - regardless of whether that reference is actually correct or not). This, however, has the tacit limitation that the accuracy of a system developed using machine learning techniques is only as good as the reference sensor or 'source of truth' used to train it.

10.4 Cost Analysis



Description	Supplier	Unit Price [ZAR]	Qty	Total [ZAR]
6061-T4 aluminium bottom plate	Fabrinox	148,56	1	148,56
6061-T4 aluminium top plate	Fabrinox	102,76	1	102,76
Machined Aluminium mounting brackets*	UCT	400,00	8	3200,00
TAS501M miniature s-type load cell**	AliExpress	283,00	4	1132,00
STM32F407 MCU	Digi-Key	188,30	1	188,30
OpenLog data logger	Bot Shop	104,65	1	104,65
8GB class 10 micro SD card	Micro Robotics	115,00	1	115,00
AD7730	Digi-Key	209,13	4	836,52
Auxiliary plugs, cables and passive components	Digi-Key	50,00	1	50,00
Momentary push switches 50mA	Mantech	7,41	26	192,66
Neoprene rubber top plate cover	Rubber Products & Mouldings	30,00	1	30,00
Total [ZAR]				6100,45

Figure 10.2: Table showing the approximate cost breakdown of the force platform and data acquisition system.

As seen in Table 10.2 above, this entire prototype system can be implemented for approximately R6100.45. The actual price of the machined aluminium mounts is unknown as these parts were machined by the UCT Mechanical Engineering Workshop for free in this project. Based on opinions from people in the UCT Mechatronics Lab with experience in this area, the cost of the material and machining was estimated to be R400 per mount at an external company. Furthermore, the price of the load cells was converted from a base price of $20 (USD) assuming a R14/$ exchange rate.

This prototype is approximately 50 times cheaper than the AMTI BP400600-1000 industrial force plate documented in Section 2.5.3. Furthermore, the BP400600-1000 weights 32kg and the platform developed in this system only weights approximately 8kg making it significantly more portable. Although, unlike the BP400600-1000, this system is not yet capable of estimating 3-axis moments, the majority of studies [9, 10, 2, 1, 8] documented in Section 2 only utilised 3-axis *forces* in their experiments (and often, only two of the force components).

Chapter 11

Conclusions and Future Work

Over the course of this project, a low-cost force platform capable of accurately estimating 3-axis GRFs was developed. This project has successfully addressed the research questions posed in Section 1.4:

1. A low noise, high fidelity data acquisition system capable of sampling 1kHz can be developed using relatively low-cost components
2. It is possible to accurately estimate 3-axis GRFs with only four single-axis load cells using machine learning and the mechanical design outlined in Section 5.1. This can be achieved for a few thousand rands (just over R6000).
3. It is foreseeable that this force plate system, with refinement and further testing, could present a viable low-cost alternative to high end industrial versions used in the commercial and academic study of humans and animals

As can be seen by the results in Chapter 9 that were discussed in Chapter 10, the use of machine learning techniques presented a highly effective means of interpreting the non-linear relationship created by the novel mechanical structure proposed in Section 5.1. Impressively, the optimum performance of this novel system in estimating 3-axis GRFs showed smaller RMS and maximum errors (for all three component axes) than the MIT study documented in Section 2.4.2 for comparable tests.

Results in Section 9.4.3 showed that the system developed was capable of estimating both fixed loads and highly impulsive 3-axis force profiles induced during natural human gait (both for walking and light running). The results obtained during these experiments were validated by findings in [9] and [11].

The electronic hardware and embedded software aspects comprising data acquisition system developed for sampling at 1kHz proved extremely effective and consistent once operational. It is believed that the fidelity and resolution of data acquired from it will only improve with the use of the PCBs designed in Section 6.2.3. Data integrity techniques implemented, as discussed in Section 7.3.1, proved invaluable in detecting communication errors and preventing erroneous data reaching the machine learning models.

To date, no low-cost[1] force platforms capable of accurately estimating 3-axis GRFs exist. The platform developed in this project presents an extremely promising and exciting low cost proof-of-concept model that, with further refinement and testing, could soon pose a feasible alternative to commercial force plates used in industry (such as the BP400600 and equivalent models).

11.1 Recommendations for Future Work

While the system performed remarkably well in fulfilling the objectives of this project, there are many opportunities for refinement of the system and experimental procedure followed given more time. Recommendations for future work are presented below.

Augmented models and improved parameter selection

It could prove useful to conduct a more detailed analysis of the mechanical constraints and relationships in the system to determine whether even a basic model capturing some of the system dynamics can be developed. This could be used in model-based machine learning techniques which would offer more transparency in their functionality (unlike with a neural network which encodes relationships in its thousands of weights and biases) and would be less dependent on large volumes of training data.

Methods such as k-fold cross validation should be investigated to optimise model parameters for generalisation across data sets. Furthermore, considering the negligible time taken to train the ELM models used in this project, an ensemble of ELMs could be implemented to produce an estimated output as the majority vote of a strategically chosen subset of independent ELM models. This would exploit the diversity introduced by the inherent randomness in the ELM's hidden weights.

Investigating continuous xy sensing and training mechanisms

A continuous xy sensing mechanism to determine the COP of the applied load should be investigated. Proprietary mechanisms for achieving this through capacitive or resistive sensing elements would be ideal. However, existing products such as the SparkFun SoftPot membrane potentiometer are thin strips that change resistance linearly along their length when subjected to a force. These may provide a means of achieving a near-continuous sensing method if many

[1] in the region of a few thousand rands

strips are placed across the plate. Capacitive films (similar to those used in touch screens) should also be investigated.

Figure 11.1: Image of the SparkFun SoftPot membrane potentiometer [57]

A training mechanism that could provide consistent, linearly increasing loads across a continuous xy grid would be ideal and would expose models to loads across the whole plate, instead of at 26 discrete positions as in this project. This was achieved by using a CNC machine in the MIT study in [26]. This would also allow more force to be applied steadily over a wider range and would allow the true minimum and maximum force estimation limits of the system to be ascertained.

Optimising the mechanical design

It would be useful to investigate the effect of modifying the machined load cell mounts to secure them at angles other than 45°. For example, mounting at a more oblique angle may allow for greater sensing resolution in the lateral axes.

Cost considerations

Theoretically, the STM32F0 MCU variants (Cortex-M0 core) could fulfil the same purpose as the STM32F4 used in this project since the STM32F051C6, for example, meets the hardware peripheral requirements of 1 DMA, 2 SPI and 2 UART peripherals and as listed by supplier Digi-Key, only costs R40 compared to R188. However, this MCU cost is not of great concern since the major cost of this system as shown in Table 10.2 arose from the machining of the 8 aluminium load cell mounts. A cheaper material capable of withstanding the expected mechanical loads should be investigated in future work. Particularly one that lends itself to moulding or cheaper manufacturing methods.

Extended training and testing

Given the data-driven nature of the system, it would benefit greatly from more training data. Human gait experiments should be explored more extensively in future work with subjects of varying mass and physical characteristics. Gathering data from experiments with trained dogs would also provide useful training data leading up to experiments to be performed with the cheetah.

Future functionality

As the OptoForce sensor used to measure actual GRFs was unable to measure torques, 3-axis moments could not be considered in this project. Future research could include using a more advanced F/T sensor capable of measuring 3-axis torques in addition to GRFs to investigate this system's ability to estimate 3-axis moments (using the existing GRF estimation capability and some basic mathematics).

As it is envisaged by the UCT Mechatronics Lab that several low-cost force plates will be arranged in a fashion similar to that in Figure 2.5, a means of synchronising the plates should be investigated in future research. This may potentially be achieved through physically connecting and controlling the $\overline{\text{SYNC}}$ pins of the AD7730 chips across the plates or implementing a wireless synchronisation system through Bluetooth.

Bibliography

[1] T. Suchomel, "Force Plate Use in Performance Monitoring and Sport Science Testing", *IAAF*, 2014.

[2] A. Coetsee, "Analysis of the Vertical Ground Reaction Forces in Sports Participants with Abductor-related Groin Pain: a Comparison Study," Msc Thesis, University of Stellenbosch, Stellenbosch, 2016

[3] D. Ortega, R. Bes, E. C. R. de la Rosa, J. Francisco, "Analysis of the vertical ground reaction forces and temporal factors in the landing phase of a countermovement jump," *Journal of Sports Science & Medicine*, 2010, 9(2), 282.

[4] P. J. McNair, H. Prapavessis & K. Callender, "Decreasing landing forces: Effect of instruction," *British Journal of Sports Medicine*, 2000, 34(4), 293-296.

[5] T. E. Hewett, "Biomechanical measures of neuromuscular control and valgus loading of the knee predict anterior cruciate ligament injury risk in female athletes: A prospective study," *The American Journal of Sports Medicine*, 2000, 33(4), 492-501.

[6] F. Impellizzeri, E. Rampinini and S. Marcora, "Physiological assessment of aerobic training in soccer," *Journal of Sports Sciences*, vol. 23, no. 6, pp. 583-592, 2005.

[7] F. Loffing, N. Hagemann, B. Strauss and C. MacMahon, *Laterality in sports*, Academic Press, 2012.

[8] L. Prosperini and C. Pozzilli, "The Clinical Relevance of Force Platform Measures in Multiple Sclerosis: A Review," *Multiple Sclerosis International*, vol. 2013, pp. 19, 2013.

[9] H. H. Manap and N. M. Tahir, "Detection of Parkinson gait pattern based on vertical ground reaction force," in *2013 IEEE International Conference on Control System, Computing and Engineering*, 2013.

[10] M. E. Morris, R. Iansek, T. A. Matyas, and J. J. Summers, "Stride length regulation in Parkinsons disease," *Brain*, vol. 119, no. 2, pp. 551568, 1996.

[11] P. G. Rosquist et al., "Estimation of 3D Ground Reaction Force Using Nanocomposite Piezo-Responsive Foam Sensors During Walking," *Annals of Biomedical Engineering*, vol. 45, no. 9, pp. 21222134, May 2017.

[12] R. LeMoyne, T. Mastroianni, A. Hessel, and K. Nishikawa, "Application of a Multilayer Perceptron Neural Network for Classifying Software Platforms of a Powered Prosthesis through a Force Plate," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 2015.

[13] E. Brown, "Cheetah III robot preps for a role as a first responder," *MIT News*, 2018. [Online]. Available: http://news.mit.edu/2018/cheetah-robot-preps-role-first-responder-sangbae-kim-0326.

[14] "WildCat - The World's Fastest Quadruped Robot," *Bostondynamics.com*, 2018. [Online]. Available:https://www.bostondynamics.com/wildcat.

[15] G. Liu, H. Lin, H. Lin, S. Chen and P. Lin, "A Bio-Inspired Hopping Kangaroo Robot with an Active Tail," *Journal of Bionic Engineering*, vol. 11, no. 4, pp. 541-555, 2014.

[16] S. McCarty, "Impact Force Patterns on a Landing Cat," *ISB Journal of Science*, vol. 8, no. 1, 2014.

[17] A. Patel, B. Stocks, C. Fisher, F. Nicolls, and E. Boje, "Tracking the Cheetah Tail Using Animal-Borne Cameras, GPS, and an IMU," *IEEE sensors letters*, vol. 1, pp. 1-4, 2017.

[18] A. Patel, C. Fisher, B. Stocks, F. Nicolls, and E. Boje, "Tracking the Cheetah Tail and Spine using Animal-borne Cameras and a Wireless Sensor Network," in *Integrative and Comparative Biology*, 2017, pp. E373-E373.

[19] A. Patel, E. Boje, C. Fisher, L. Louis, and E. Lane, "Quasi-steady state aerodynamics of the cheetah tail,"*Biology Open*, vol. 5, pp. 1072-1076, 2016.

[20] A. Patel and M. Braae, "Rapid turning at high-speed: Inspirations from the cheetah's tail," in *Intelligent Robots and Systems (IROS)*, 2013 IEEE/RSJ International Conference, 2013, pp. 5506-5511

[21] M. Hildebrand, "Motions of the running cheetah and horse," *Journal of Mammalogy*, vol. 40, no. 4, p. 481, 1959.

[22] P. Hudson, S. Corr and A. Wilson, "High speed galloping in the cheetah (Acinonyx jubatus) and the racing greyhound (Canis familiaris): spatio-temporal and kinetic characteristics," *Journal of Experimental Biology*, vol. 215, no. 14, pp. 2425-2434, 2012.

[23] Y. Guo, "A New Proxy Measurement Algorithm with Application to the Estimation of Vertical Ground Reaction Forces Using Wearable Sensors," *Sensors*, vol. 17, no. 10, p. 2181, Sep. 2017.

[24] A. Karatsidis, G. Bellusci, H. Schepers, M. de Zee, M. Andersen, and P. Veltink, "Estimation of Ground Reaction Forces and Moments During Gait Using Only Inertial Motion Capture," *Sensors*, vol. 17, no. 12, p. 75, Dec. 2016.

[25] A. W. Garcia, C. R. Langenthal, R. M. Angulo-Barroso, and M. M. Gross, "A Comparison of Accelerometers for Predicting Energy Expenditure and Vertical Ground Reaction Force in School-Age Children," *Measurement in Physical Education and Exercise Science*, vol. 8, no. 3, pp. 119144, Sep. 2004.

[26] S. Chuah, Meng YeeKim, "Enabling force sensing during ground locomotion: A bio-inspired, multi-axis, composite force sensor using discrete pressure mapping," *IEEE Sensors*, vol. 14, no. 5, pp. 16931703, 2014. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6714415

[27] D. Alveringh, R. A. Brookhuis, "27th IEEE International Conference on Micro Electro Mechanical Systems, MEMS," p. 680-683 4 p, 2014

[28] D. Kim, "Six-axis Capacitive Force/Torque Sensor Based on Dielectric Elastomer," in *SPIE - The International Society for Optical Engineering*, 86872J.

[29] J. Huang, C. Y. Wong, "Design of a Novel Six-Axis Force/Torque Sensor based on Optical Fibre Sensing for Robotic Applications," in *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics*, SCITEPRESS - Science and Technology Publications.

[30] ATI Industrial Automation, "F/T Sensor: Axia80," Axia80 datasheet.

[31] "Hall Effect and Strain Gage Sensing Technology for Multi-axis Force Plates and Force Sensors," Amti.biz, 2018. [Online]. Available: https://www.amti.biz/fps-sensor-tech2.aspx.

[32] Vernier Software and Technology, "Force Plate," FP-BTA datasheet.

[33] T. Kleckers, "How does a Piezoelectric Force Transducer Work?" *HBM*, 2018. [Online]. Available: https://www.hbm.com/en/7318/how-does-a-piezoelectric-force-transducer-work/

[34] H. R. Kamrul, "Design and Construction of a Strain Gauge," 2016.

[35] National Instruments, "Measuring Strain with Strain Gages," *Ni.com*, 2016. [Online]. Available: http://www.ni.com/white-paper/3642/en/.

[36] "Force plates - do you need one?," *Kinetic.com.au*, 2016. [Online]. Available:https://kinetic.com.au/pdf/Force_Plates.pdf

[37] "Linear regression", En.wikipedia.org, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Linear_regression.

[38] C. Cortes and V. Vapnik, *Machine Learning*, vol. 20, no. 3, pp. 273297, 1995.

[39] A. Smola and B. Schlkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199-222, 2004.

[40] J. Greene, "Computational Intelligence," UCT Course Notes, University of Cape Town, Cape Town, 2018.

[41] E. Pienaar, "Analytics Part 2: Introduction to Neural Networks," 1st ed., University of Cape Town, Cape Town, 2018.

[42] M. Nielsen, *Neural Networks and Deep Learning*, Determination Press, 2015.

[43] "Underfitting vs. Overfitting, scikit-learn 0.20.0 documentation," *Scikit-learn.org*, 2018. [Online]. Available at:http://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html

[44] M. D. Tissera and M. D. McDonnell, "Deep extreme learning machines: supervised autoencoding architecture for classification," *Neurocomputing*, vol. 174, pp. 4249, Jan. 2016.

[45] G. Huang, H. Zhou and X. Ding, "Extreme Learning Machine for Regression and Multiclass Classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513-529, 2012.

[46] G. Huang, Q. Zhu and C. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489-501, 2006.

[47] "6061 aluminium alloy," *En.wikipedia.org*. [online] Available at: https://en.wikipedia.org/wiki/6061_aluminium_alloy.

[48] OptoForce, "3-Axis Force Sensor," OMD-45-FH-2000N datasheet.

[49] Analog Devices, "Bridge Transducer ADC," AD7730/AD7730L datasheet, 2012.

[50] AVIA Semiconductor, "24-bit ADC for Weigh Scales," HX711 datasheet.

[51] W. Chou, T. H. Meng, "Time domain analysis of sigma delta modulation," *Acoustics, Speech, and Signal Processing*, 1990.

[52] C. Noviello, "Mastering STM32," Leanpub.

[53] Atmel, "8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash," ATmega328p datasheet.

[54] STMicroelectronics, "STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm-based 32-bit MCUs," Reference manual RM0090.

[55] STMicroelectronics, "Using the CRC peripheral in the STM32 family," Appl. Note AN4187

[56] "Cyclic redundancy check", *En.wikipedia.org*, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Cyclic_redundancy_check.

[57] "SoftPot Membrane Potentiometer," *Sparkfun.com*, 2018. [Online]. Available: https://www.sparkfun.com/products/8681.

[58] "Micro SD Card Breakout Board", *Learn.adafruit.com*, 2018. [Online]. Available: https://learn.adafruit.com/adafruit-micro-sd-breakout-board-card-tutorial/introduction.

[59] [11]"SparkFun OpenLog", *Sparkfun.com*, 2018. [Online]. Available: https://www.sparkfun.com/products/13712.

[60] Y. Kwon, "Ground Reaction Force," *kwon3d.com.* [Online]. Available at: http://www.kwon3d.com/theory/grf/grf.html.

[61] E. Rensch, "What kind of environment does a cheetah live in?" *Sciencing.com.* [Online]. Available at: https://sciencing.com/kind-environment-cheetah-live-8604570.html.

Appendix A

Positional Testing: Explanatory Diagrams and Results

A.1 Switch Co-ordinate System

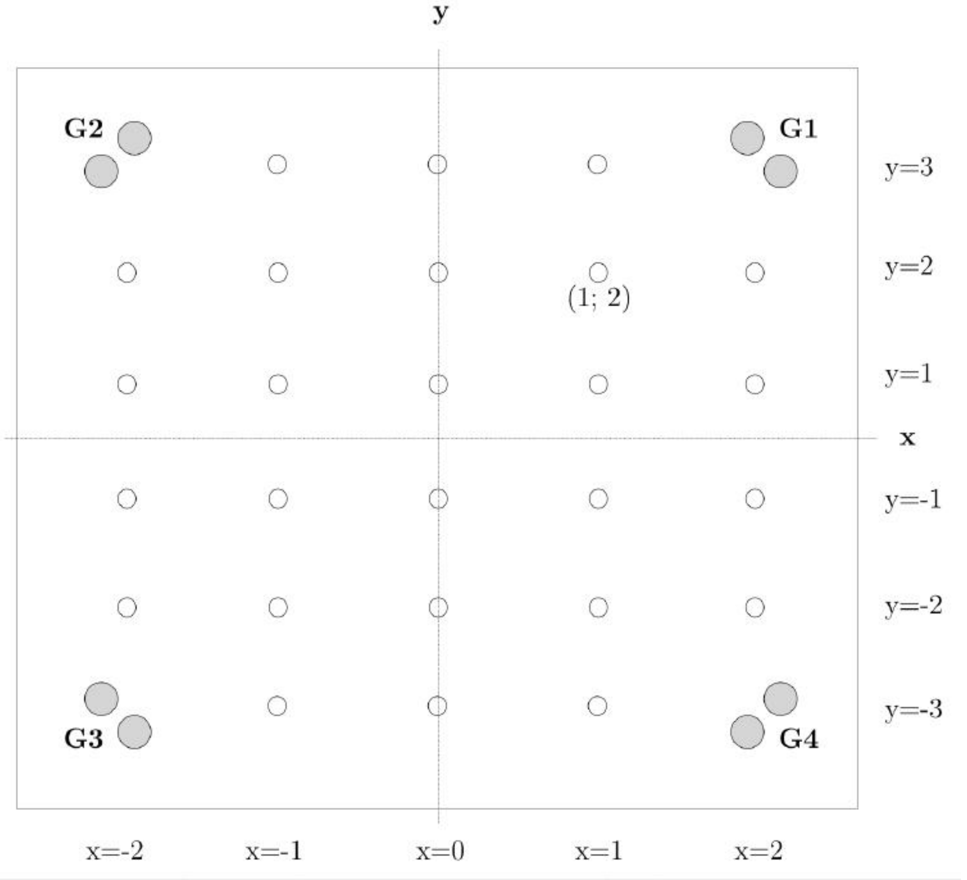


Figure A.1: Diagram defining the the plate co-ordinate system used in experiments. The hollow circles represent the 26 switch positions and the grey circles depict the mounting bolt positions for reference. An example switch at position (1;2) is labelled.

A.2 Single Axis Positional Testing Results

The resulting prediction errors for single axis (vertical force) data sets obtained over the entire grid of 26 points are presented in tables A.1 to A.3 below. Note that all results were obtained using completely unseen (out-of-sample) data that was not used in training of any of the regression models. **R** represents the radius of a particular point from the *xy* origin (geometric centre) of the plate in cm and **x** and **y** are the co-ordinates of the COP position of the force applied. Note that these are *dimensionless* units. Note that all regression models referred to below, unless otherwise specified, were implemented with the default architecture and parameters defined in Section 8.1.4.

Table A.1: Table showing the Neural Network prediction errors for vertical force (single axis) positional tests.

x	**y**	**R [cm]**	**RMSE [N]**	**Error SD [N]**	**Max Error [N]**	RMSE [%FS]	**Error SD [%FS]**	**Max Error [%FS]**
1	3	9.4663	6.8957	10.959	28.51	**0.34478**	0.54795	1.4255
0	3	8.5	8.3898	14.067	39.586	**0.41949**	0.70334	1.9793
-1	3	9.4663	8.8129	11.532	28.557	**0.44065**	0.5766	1.4278
2	2	9.7701	9.3202	12.673	37.044	**0.46601**	0.63365	1.8522
1	2	6.5857	6.5813	9.9551	25.931	**0.32906**	0.49775	1.2965
0	2	5.1	6.2481	8.7465	23.091	**0.31241**	0.43733	1.1545
-1	2	6.5857	8.4211	13.316	33.321	**0.42106**	0.66582	1.666
-2	2	9.7701	7.6246	11.906	36.198	**0.38123**	0.59531	1.8099
2	1	8.505	10.576	11.298	33.177	**0.52878**	0.56492	1.6589
1	1	4.5001	9.3501	12.614	33.762	**0.46751**	0.63072	1.6881
0	1	1.7	6.2816	8.8505	23.179	**0.31408**	0.44252	1.1589
-1	1	4.5001	7.2674	10.743	27.026	**0.36337**	0.53717	1.3513
-2	1	8.505	6.7116	10.925	31.693	**0.33558**	0.54626	1.5846
2	-1	8.505	7.298	12.622	38.165	**0.3649**	0.6311	1.9082
1	-1	4.5001	8.1744	11.318	26.028	**0.40872**	0.56589	1.3014
0	-1	1.7	6.2816	8.8505	23.179	**0.31408**	0.44252	1.1589
-1	-1	4.5001	7.3496	11.195	28.485	**0.36748**	0.55973	1.4243
-2	-1	8.505	5.7986	9.9442	27.187	**0.28993**	0.49721	1.3594
2	-2	9.7701	5.3888	7.8371	21.912	**0.26944**	0.39186	1.0956
1	-2	6.5857	6.6683	9.9942	26.656	**0.33341**	0.49971	1.3328
0	-2	5.1	6.3151	8.9544	23.267	**0.31576**	0.44772	1.1633
-1	-2	6.5857	7.1991	12.281	32.358	**0.35995**	0.61406	1.6179
-2	-2	9.7701	9.5327	13.576	35.264	**0.47664**	0.6788	1.7632
1	-3	9.4663	6.1826	10.809	29.288	**0.30913**	0.54043	1.4644
0	-3	8.5	9.6835	14.765	68.13	**0.48418**	0.73826	3.4065
-1	-3	9.4663	10.31	13.217	32.508	**0.51548**	0.66084	1.6254

Table A.2: Table showing the linear regressor prediction errors for vertical force (single axis) positional tests.

x	**y**	**R [cm]**	**RMSE [N]**	**Error SD [N]**	**Max Error [N]**	**RMSE [%FS]**	**Error SD [%FS]**	**Max Error [%FS]**
1	3	9.4663	10.802	12.209	28.049	**0.5401**	0.61044	1.4025
0	3	8.5	8.4578	13.202	36.693	**0.42289**	0.66011	1.8346
-1	3	9.4663	11.249	11.216	29.998	**0.56246**	0.5608	1.4999
2	2	9.7701	12.544	14.158	37.352	**0.6272**	0.7079	1.8676
1	2	6.5857	7.0253	8.993	21.014	**0.35126**	0.44965	1.0507
0	2	5.1	6.6698	8.7121	19.901	**0.33349**	0.4356	0.99506
-1	2	6.5857	10.392	16.393	43.127	**0.51959**	0.81967	2.1564
-2	2	9.7701	9.3931	11.967	30.517	**0.46965**	0.59836	1.5258
2	1	8.505	8.6637	11.784	33.536	**0.43318**	0.5892	1.6768
1	1	4.5001	10.597	17.968	41.616	**0.52986**	0.8984	2.0808
0	1	1.7	6.8094	8.6785	20.592	**0.34047**	0.43392	1.0296
-1	1	4.5001	10.662	16.25	41.286	**0.53312**	0.8125	2.0643
-2	1	8.505	9.8341	11.834	29.933	**0.4917**	0.59172	1.4966
2	-1	8.505	12.077	16.75	36.761	**0.60383**	0.8375	1.8381
1	-1	4.5001	8.6514	11.519	25.726	**0.43257**	0.57597	1.2863
0	-1	1.7	6.8094	8.6785	20.592	**0.34047**	0.43392	1.0296
-1	-1	4.5001	13.412	17.17	39.71	**0.67058**	0.85849	1.9855
-2	-1	8.505	10.275	11.702	29.349	**0.51376**	0.58508	1.4674
2	-2	9.7701	9.243	11.462	32.211	**0.46215**	0.57312	1.6105
1	-2	6.5857	7.8592	10.62	29.804	**0.39296**	0.53102	1.4902
0	-2	5.1	6.9489	8.6449	21.282	**0.34745**	0.43225	1.0641
-1	-2	6.5857	8.9905	13.736	35.305	**0.44953**	0.68682	1.7653
-2	-2	9.7701	11.157	15.892	42.92	**0.55785**	0.79459	2.146
1	-3	9.4663	7.0039	9.4273	24.224	**0.3502**	0.47137	1.2112
0	-3	8.5	10.376	15.73	74.89	**0.51878**	0.78652	3.7445
-1	-3	9.4663	14.94	18.744	46.071	**0.747**	0.93719	2.3036

Table A.3: Table showing the ELM prediction errors for vertical force (single axis) positional tests.

x	**y**	**R [cm]**	**RMSE [N]**	**Error SD [N]**	**Max Error [N]**	RMSE [%FS]	**Error SD [%FS]**	**Max Error [%FS]**
1	3	9.4663	7.9825	11.023	29.099	**0.39913**	0.55113	1.4549
0	3	8.5	9.2163	15.038	41.338	**0.46082**	0.75192	2.0669
-1	3	9.4663	8.2212	12.445	32.713	**0.41106**	0.62223	1.6356
2	2	9.7701	10.998	14.65	45.414	**0.54992**	0.73249	2.2707
1	2	6.5857	7.5583	9.7281	23.003	**0.37791**	0.4864	1.1501
0	2	5.1	6.6727	9.481	24.313	**0.33364**	0.47405	1.2156
-1	2	6.5857	9.8502	14.285	33.376	**0.49251**	0.71425	1.6688
-2	2	9.7701	8.0922	12.685	39.194	**0.40461**	0.63423	1.9597
2	1	8.505	11.305	11.262	32.971	**0.56524**	0.56312	1.6485
1	1	4.5001	9.6807	12.692	34.668	**0.48404**	0.63462	1.7334
0	1	1.7	7.4223	10.517	24.649	**0.37111**	0.52583	1.2325
-1	1	4.5001	7.4794	12.233	32.297	**0.37397**	0.61165	1.6148
-2	1	8.505	7.8157	12.571	37.346	**0.39078**	0.62855	1.8673
2	-1	8.505	7.4752	12.417	36.689	**0.37376**	0.62085	1.8344
1	-1	4.5001	8.6408	11.65	26.959	**0.43204**	0.5825	1.348
0	-1	1.7	7.4223	10.517	24.649	**0.37111**	0.52583	1.2325
-1	-1	4.5001	7.2586	10.997	26.348	**0.36293**	0.54984	1.3174
-2	-1	8.505	7.5392	12.457	35.498	**0.37696**	0.62287	1.7749
2	-2	9.7701	5.5324	8.3174	22.654	**0.27662**	0.41587	1.1327
1	-2	6.5857	6.6263	10.737	30.361	**0.33132**	0.53684	1.5181
0	-2	5.1	8.1719	11.552	24.986	**0.40859**	0.57762	1.2493
-1	-2	6.5857	7.4513	12.519	33.8	**0.37257**	0.62594	1.69
-2	-2	9.7701	10.233	14.549	36.479	**0.51167**	0.72743	1.824
1	-3	9.4663	6.7387	9.221	23.79	**0.33694**	0.46105	1.1895
0	-3	8.5	24.251	20.696	55.432	**1.2126**	1.0348	2.7716
-1	-3	9.4663	8.7451	12.782	33.746	**0.43725**	0.6391	1.6873

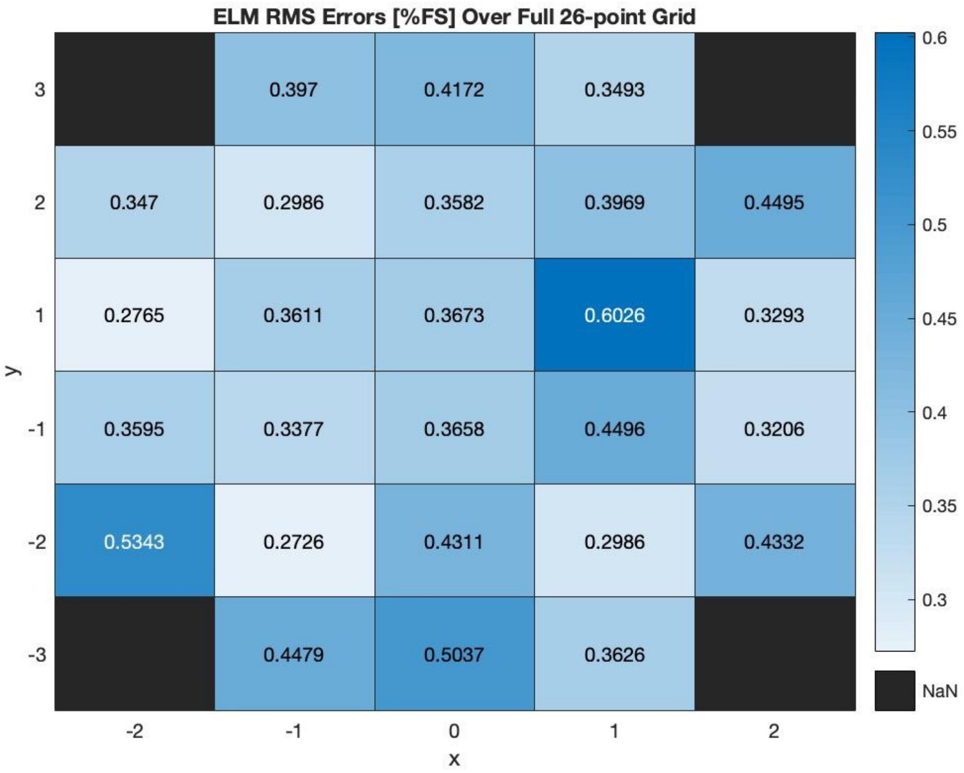


Figure A.2: Heat map over the 26 switch positions showing the RMS errors of an ELM predictor as a percentage of the full scale force of 2000N.

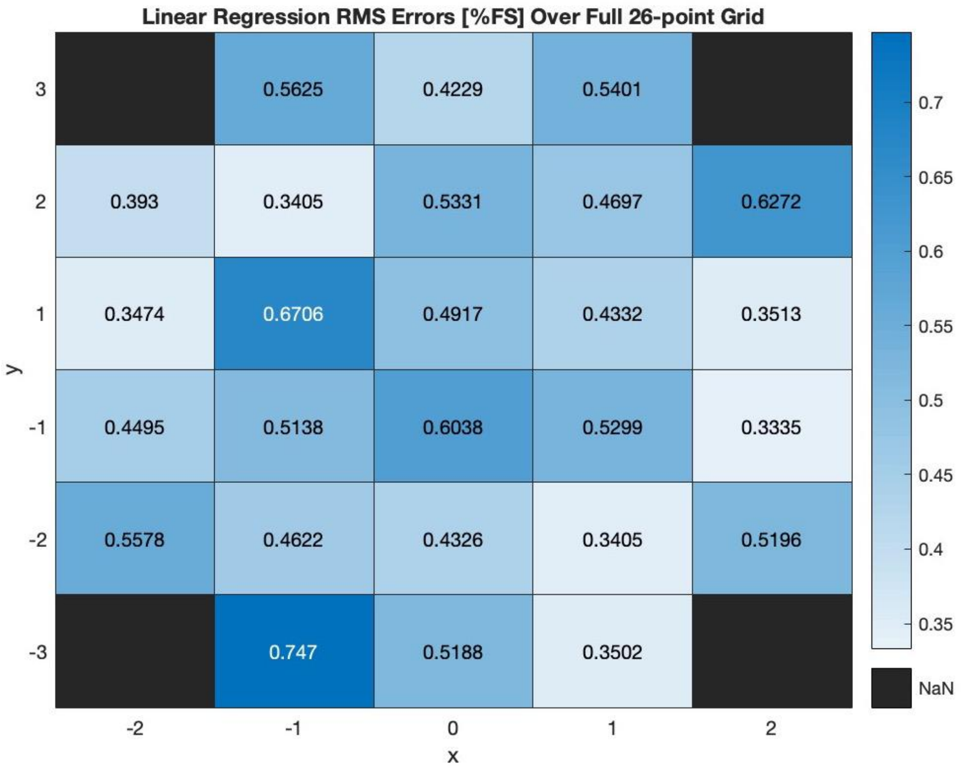
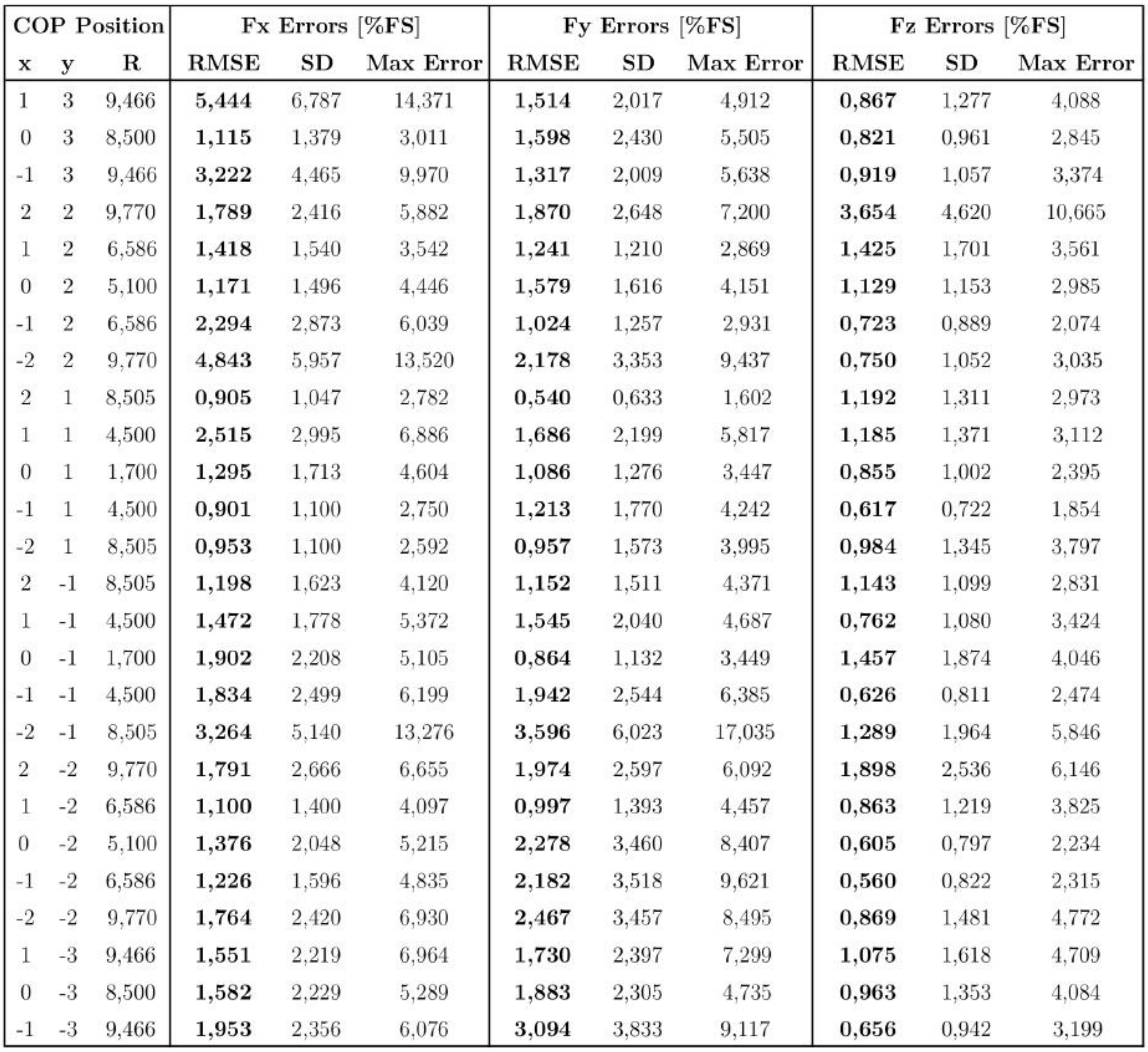


Figure A.3: Heat map over the 26 switch positions showing the RMS errors of a linear regression predictor as a percentage of the full scale force of 2000N.

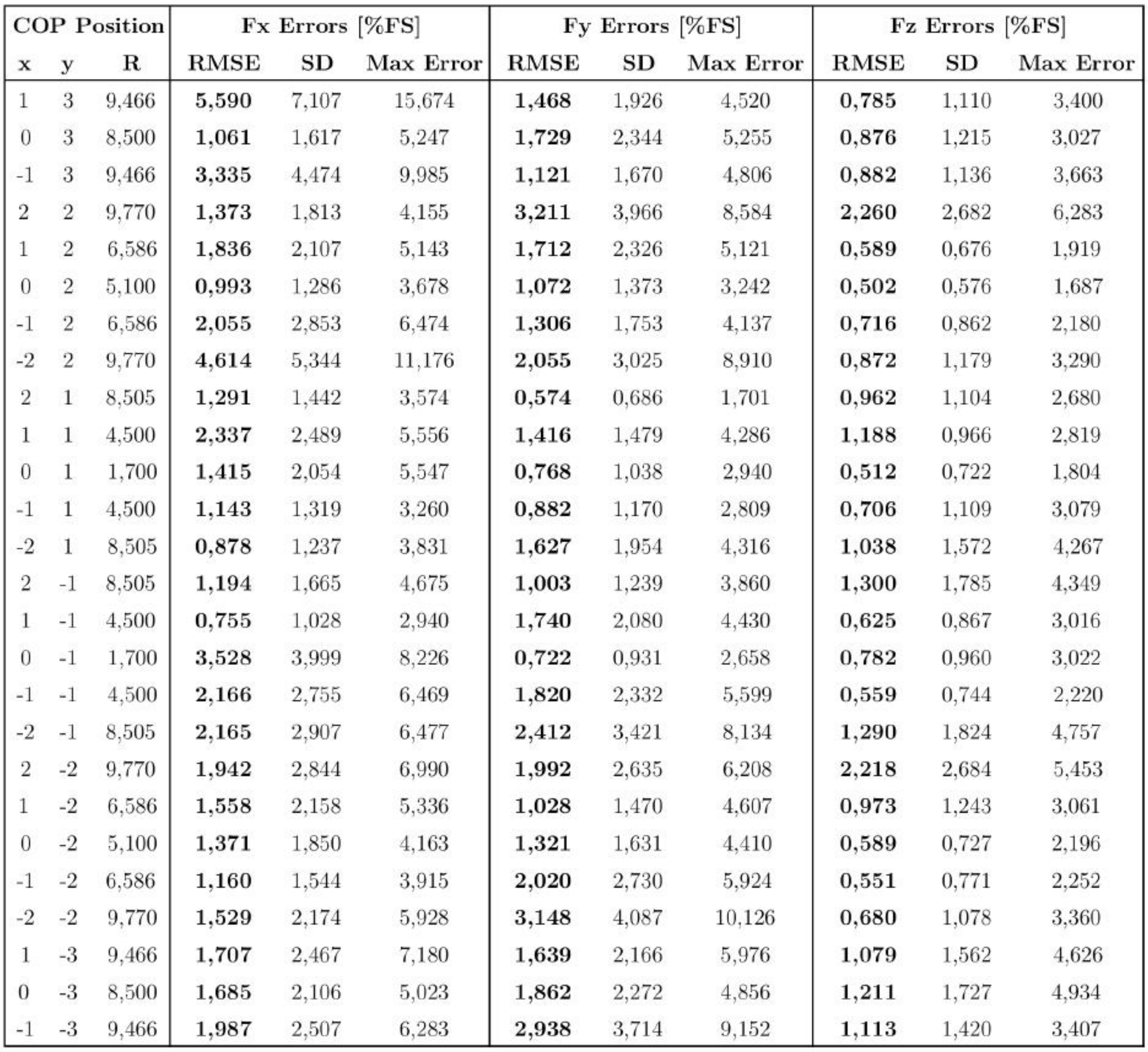
A.3 3 Axis Positional Testing Results

The resulting prediction errors for 3 axis experiments obtained over the entire grid of 26 points are presented in tables A.4 to A.5 below. Note that all results were obtained using completely unseen (out-of-sample) data that was not used in training of any of the regression models. As above, **R** represents the radius of a particular point from the *xy* origin (geometric centre) of the plate in centimetres and **x** and **y** are the co-ordinates of the COP position of the force applied. Note that these are *dimensionless* units. Note that all regression models referred to below, unless otherwise specified, were implemented with the default architecture and parameters defined in Section 8.1.4



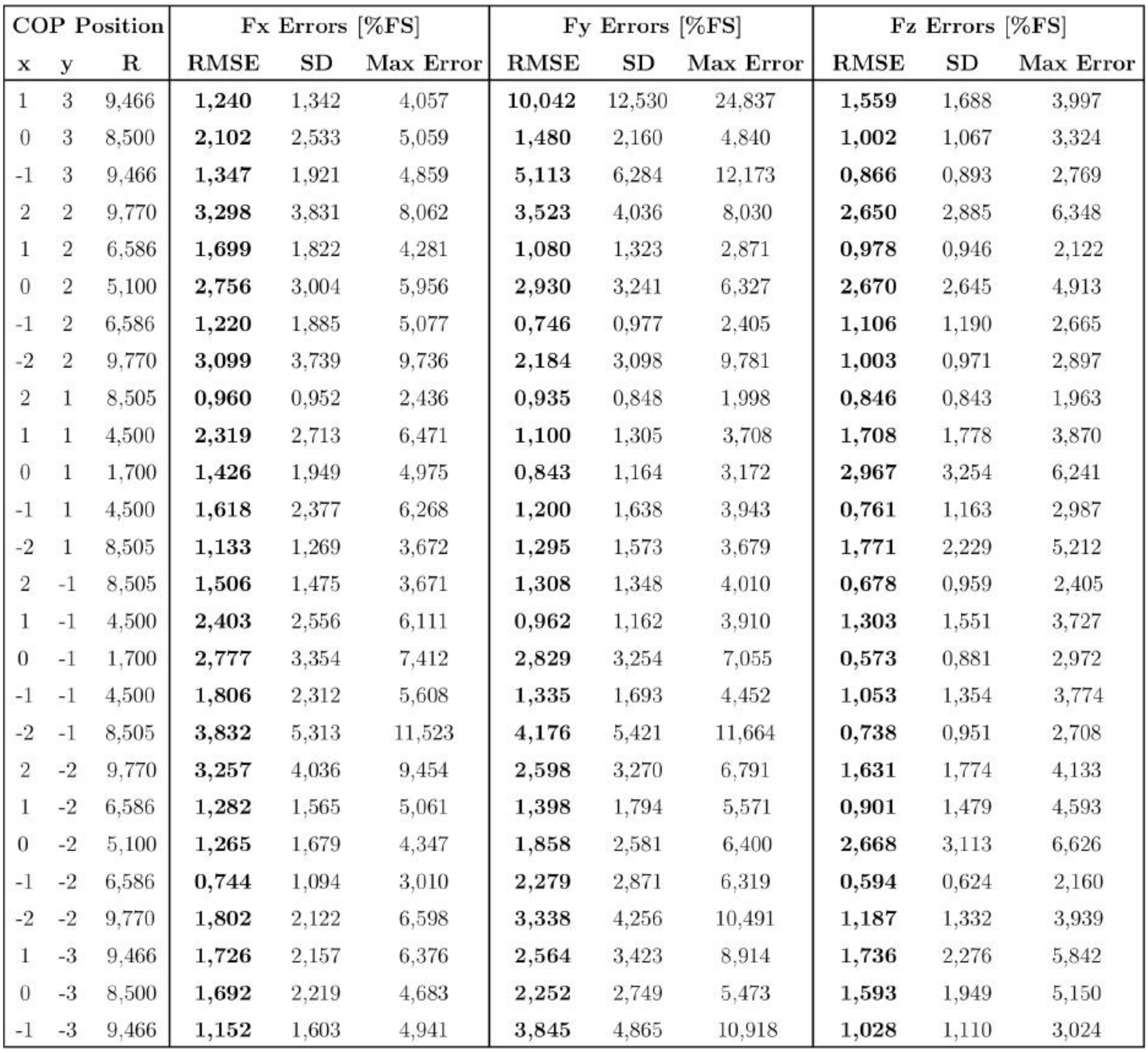
COP Position			Fx Errors [%FS]			Fy Errors [%FS]			Fz Errors [%FS]		
x	y	R	RMSE	SD	Max Error	RMSE	SD	Max Error	RMSE	SD	Max Error
1	3	9,466	**5,444**	6,787	14,371	**1,514**	2,017	4,912	**0,867**	1,277	4,088
0	3	8,500	**1,115**	1,379	3,011	**1,598**	2,430	5,505	**0,821**	0,961	2,845
-1	3	9,466	**3,222**	4,465	9,970	**1,317**	2,009	5,638	**0,919**	1,057	3,374
2	2	9,770	**1,789**	2,416	5,882	**1,870**	2,648	7,200	**3,654**	4,620	10,665
1	2	6,586	**1,418**	1,540	3,542	**1,241**	1,210	2,869	**1,425**	1,701	3,561
0	2	5,100	**1,171**	1,496	4,446	**1,579**	1,616	4,151	**1,129**	1,153	2,985
-1	2	6,586	**2,294**	2,873	6,039	**1,024**	1,257	2,931	**0,723**	0,889	2,074
-2	2	9,770	**4,843**	5,957	13,520	**2,178**	3,353	9,437	**0,750**	1,052	3,035
2	1	8,505	**0,905**	1,047	2,782	**0,540**	0,633	1,602	**1,192**	1,311	2,973
1	1	4,500	**2,515**	2,995	6,886	**1,686**	2,199	5,817	**1,185**	1,371	3,112
0	1	1,700	**1,295**	1,713	4,604	**1,086**	1,276	3,447	**0,855**	1,002	2,395
-1	1	4,500	**0,901**	1,100	2,750	**1,213**	1,770	4,242	**0,617**	0,722	1,854
-2	1	8,505	**0,953**	1,100	2,592	**0,957**	1,573	3,995	**0,984**	1,345	3,797
2	-1	8,505	**1,198**	1,623	4,120	**1,152**	1,511	4,371	**1,143**	1,099	2,831
1	-1	4,500	**1,472**	1,778	5,372	**1,545**	2,040	4,687	**0,762**	1,080	3,424
0	-1	1,700	**1,902**	2,208	5,105	**0,864**	1,132	3,449	**1,457**	1,874	4,046
-1	-1	4,500	**1,834**	2,499	6,199	**1,942**	2,544	6,385	**0,626**	0,811	2,474
-2	-1	8,505	**3,264**	5,140	13,276	**3,596**	6,023	17,035	**1,289**	1,964	5,846
2	-2	9,770	**1,791**	2,666	6,655	**1,974**	2,597	6,092	**1,898**	2,536	6,146
1	-2	6,586	**1,100**	1,400	4,097	**0,997**	1,393	4,457	**0,863**	1,219	3,825
0	-2	5,100	**1,376**	2,048	5,215	**2,278**	3,460	8,407	**0,605**	0,797	2,234
-1	-2	6,586	**1,226**	1,596	4,835	**2,182**	3,518	9,621	**0,560**	0,822	2,315
-2	-2	9,770	**1,764**	2,420	6,930	**2,467**	3,457	8,495	**0,869**	1,481	4,772
1	-3	9,466	**1,551**	2,219	6,964	**1,730**	2,397	7,299	**1,075**	1,618	4,709
0	-3	8,500	**1,582**	2,229	5,289	**1,883**	2,305	4,735	**0,963**	1,353	4,084
-1	-3	9,466	**1,953**	2,356	6,076	**3,094**	3,833	9,117	**0,656**	0,942	3,199

Figure A.4: Table showing neural network prediction errors for each of the 3-axis GRF component forces over the full 26-point positional grid



COP Position			Fx Errors [%FS]			Fy Errors [%FS]			Fz Errors [%FS]		
x	y	R	RMSE	SD	Max Error	RMSE	SD	Max Error	RMSE	SD	Max Error
1	3	9,466	**5,590**	7,107	15,674	**1,468**	1,926	4,520	**0,785**	1,110	3,400
0	3	8,500	**1,061**	1,617	5,247	**1,729**	2,344	5,255	**0,876**	1,215	3,027
-1	3	9,466	**3,335**	4,474	9,985	**1,121**	1,670	4,806	**0,882**	1,136	3,663
2	2	9,770	**1,373**	1,813	4,155	**3,211**	3,966	8,584	**2,260**	2,682	6,283
1	2	6,586	**1,836**	2,107	5,143	**1,712**	2,326	5,121	**0,589**	0,676	1,919
0	2	5,100	**0,993**	1,286	3,678	**1,072**	1,373	3,242	**0,502**	0,576	1,687
-1	2	6,586	**2,055**	2,853	6,474	**1,306**	1,753	4,137	**0,716**	0,862	2,180
-2	2	9,770	**4,614**	5,344	11,176	**2,055**	3,025	8,910	**0,872**	1,179	3,290
2	1	8,505	**1,291**	1,442	3,574	**0,574**	0,686	1,701	**0,962**	1,104	2,680
1	1	4,500	**2,337**	2,489	5,556	**1,416**	1,479	4,286	**1,188**	0,966	2,819
0	1	1,700	**1,415**	2,054	5,547	**0,768**	1,038	2,940	**0,512**	0,722	1,804
-1	1	4,500	**1,143**	1,319	3,260	**0,882**	1,170	2,809	**0,706**	1,109	3,079
-2	1	8,505	**0,878**	1,237	3,831	**1,627**	1,954	4,316	**1,038**	1,572	4,267
2	-1	8,505	**1,194**	1,665	4,675	**1,003**	1,239	3,860	**1,300**	1,785	4,349
1	-1	4,500	**0,755**	1,028	2,940	**1,740**	2,080	4,430	**0,625**	0,867	3,016
0	-1	1,700	**3,528**	3,999	8,226	**0,722**	0,931	2,658	**0,782**	0,960	3,022
-1	-1	4,500	**2,166**	2,755	6,469	**1,820**	2,332	5,599	**0,559**	0,744	2,220
-2	-1	8,505	**2,165**	2,907	6,477	**2,412**	3,421	8,134	**1,290**	1,824	4,757
2	-2	9,770	**1,942**	2,844	6,990	**1,992**	2,635	6,208	**2,218**	2,684	5,453
1	-2	6,586	**1,558**	2,158	5,336	**1,028**	1,470	4,607	**0,973**	1,243	3,061
0	-2	5,100	**1,371**	1,850	4,163	**1,321**	1,631	4,410	**0,589**	0,727	2,196
-1	-2	6,586	**1,160**	1,544	3,915	**2,020**	2,730	5,924	**0,551**	0,771	2,252
-2	-2	9,770	**1,529**	2,174	5,928	**3,148**	4,087	10,126	**0,680**	1,078	3,360
1	-3	9,466	**1,707**	2,467	7,180	**1,639**	2,166	5,976	**1,079**	1,562	4,626
0	-3	8,500	**1,685**	2,106	5,023	**1,862**	2,272	4,856	**1,211**	1,727	4,934
-1	-3	9,466	**1,987**	2,507	6,283	**2,938**	3,714	9,152	**1,113**	1,420	3,407

Figure A.5: Table showing ELM prediction errors for each of the 3 axis GRF component forces over the full 26-point positional grid



COP Position			Fx Errors [%FS]			Fy Errors [%FS]			Fz Errors [%FS]		
x	y	R	RMSE	SD	Max Error	RMSE	SD	Max Error	RMSE	SD	Max Error
1	3	9,466	**1,240**	1,342	4,057	**10,042**	12,530	24,837	**1,559**	1,688	3,997
0	3	8,500	**2,102**	2,533	5,059	**1,480**	2,160	4,840	**1,002**	1,067	3,324
-1	3	9,466	**1,347**	1,921	4,859	**5,113**	6,284	12,173	**0,866**	0,893	2,769
2	2	9,770	**3,298**	3,831	8,062	**3,523**	4,036	8,030	**2,650**	2,885	6,348
1	2	6,586	**1,699**	1,822	4,281	**1,080**	1,323	2,871	**0,978**	0,946	2,122
0	2	5,100	**2,756**	3,004	5,956	**2,930**	3,241	6,327	**2,670**	2,645	4,913
-1	2	6,586	**1,220**	1,885	5,077	**0,746**	0,977	2,405	**1,106**	1,190	2,665
-2	2	9,770	**3,099**	3,739	9,736	**2,184**	3,098	9,781	**1,003**	0,971	2,897
2	1	8,505	**0,960**	0,952	2,436	**0,935**	0,848	1,998	**0,846**	0,843	1,963
1	1	4,500	**2,319**	2,713	6,471	**1,100**	1,305	3,708	**1,708**	1,778	3,870
0	1	1,700	**1,426**	1,949	4,975	**0,843**	1,164	3,172	**2,967**	3,254	6,241
-1	1	4,500	**1,618**	2,377	6,268	**1,200**	1,638	3,943	**0,761**	1,163	2,987
-2	1	8,505	**1,133**	1,269	3,672	**1,295**	1,573	3,679	**1,771**	2,229	5,212
2	-1	8,505	**1,506**	1,475	3,671	**1,308**	1,348	4,010	**0,678**	0,959	2,405
1	-1	4,500	**2,403**	2,556	6,111	**0,962**	1,162	3,910	**1,303**	1,551	3,727
0	-1	1,700	**2,777**	3,354	7,412	**2,829**	3,254	7,055	**0,573**	0,881	2,972
-1	-1	4,500	**1,806**	2,312	5,608	**1,335**	1,693	4,452	**1,053**	1,354	3,774
-2	-1	8,505	**3,832**	5,313	11,523	**4,176**	5,421	11,664	**0,738**	0,951	2,708
2	-2	9,770	**3,257**	4,036	9,454	**2,598**	3,270	6,791	**1,631**	1,774	4,133
1	-2	6,586	**1,282**	1,565	5,061	**1,398**	1,794	5,571	**0,901**	1,479	4,593
0	-2	5,100	**1,265**	1,679	4,347	**1,858**	2,581	6,400	**2,668**	3,113	6,626
-1	-2	6,586	**0,744**	1,094	3,010	**2,279**	2,871	6,319	**0,594**	0,624	2,160
-2	-2	9,770	**1,802**	2,122	6,598	**3,338**	4,256	10,491	**1,187**	1,332	3,939
1	-3	9,466	**1,726**	2,157	6,376	**2,564**	3,423	8,914	**1,736**	2,276	5,842
0	-3	8,500	**1,692**	2,219	4,683	**2,252**	2,749	5,473	**1,593**	1,949	5,150
-1	-3	9,466	**1,152**	1,603	4,941	**3,845**	4,865	10,918	**1,028**	1,110	3,024

Figure A.6: Table showing linear regression model prediction errors for each of the 3-axis GRF component forces over the full 26-point positional grid

Figures A.7 to A.9 below show heat maps of the RMS errors from a neural network in estimating 3 axis GRFs. Error for the three GRF components separately as a percentage of the full scale force (%FS) over the 26 testing points on the plate. As before, $(x; y)$ co-ordinates are shown with reference to the plate co-ordinate scheme detailed in Figure 9.5. Only the heat maps correspond to the 3 axis components of the *neural network* model are shown as it was deemed that patterns exhibited by the other models did not offer any more useful information.

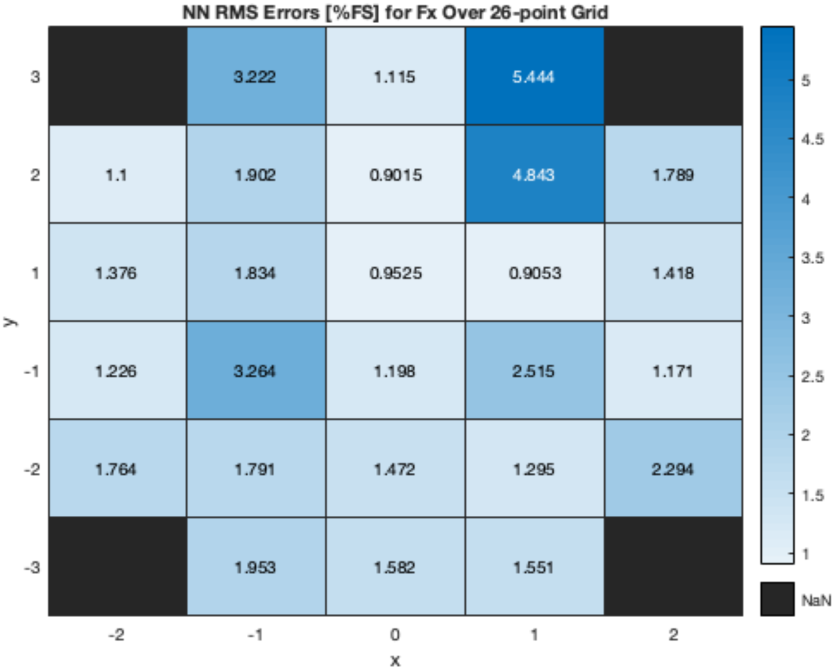


Figure A.7: Heat map over the 26 switch positions showing the RMS errors of the F_x profile estimated by a NN with default architecture

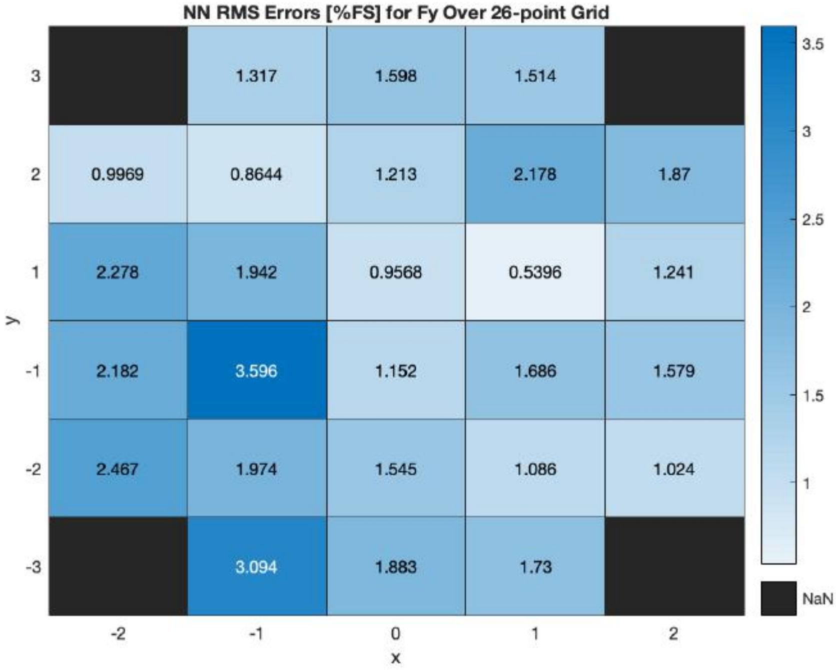


Figure A.8: Heat map over the 26 switch positions showing the RMS errors of the F_y profile estimated by a NN with default architecture

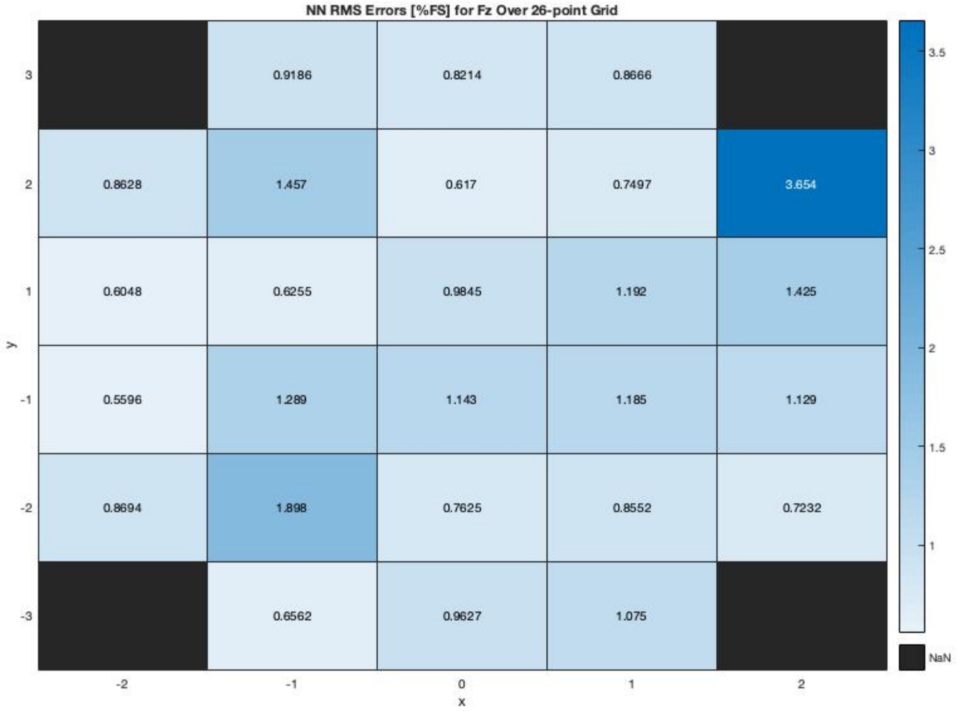


Figure A.9: Heat map over the 26 switch positions showing the RMS errors of the F_z profile estimated by a NN with default architecture

Appendix B

Mechanical Drawings

B.1 OptoForce Mounting Brace

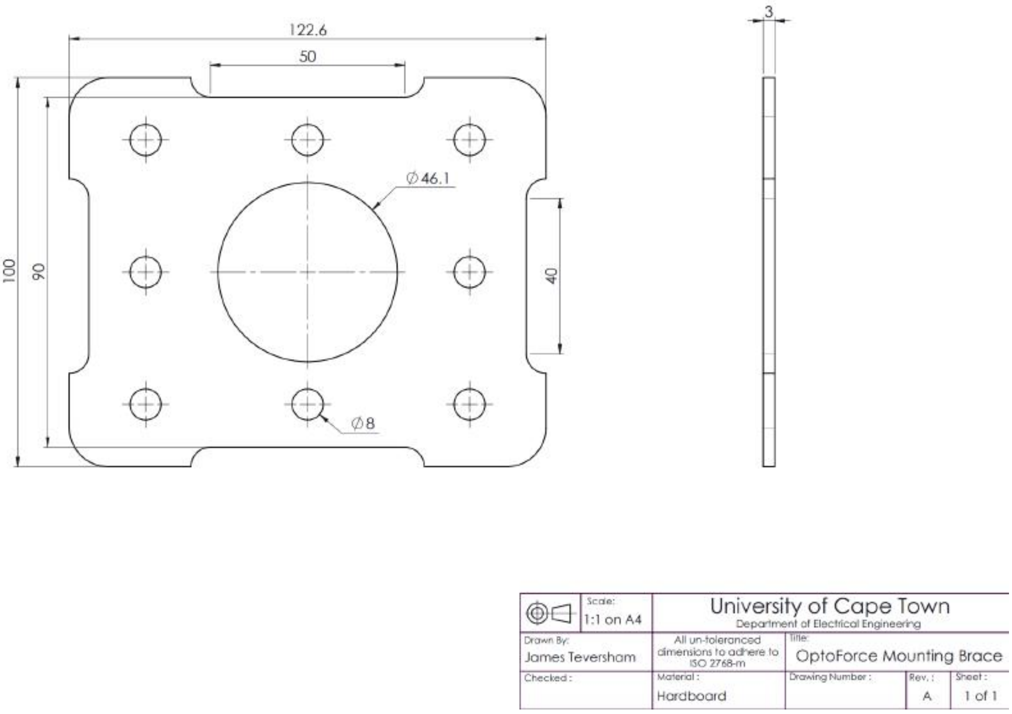


Figure B.1: Mechanical drawing of the mounting brace designed to prevent unwanted lateral motion of the OptoForce sensor during testing

B.2 Force Plate Structural Components

All components in this section (unless explicitly specified) are based on the mechanical design proposed by Patel, Fisher et al.

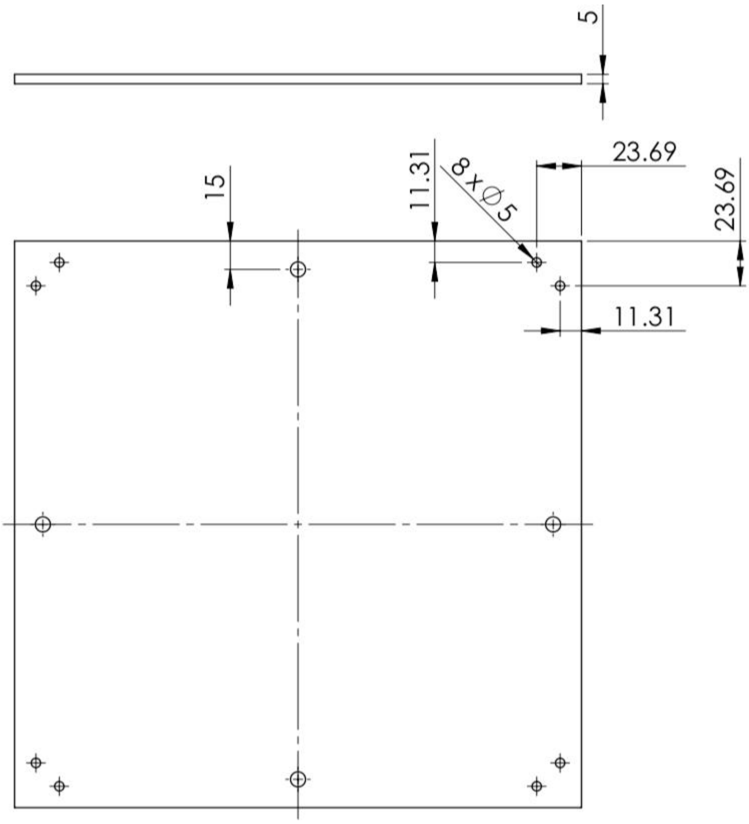


Figure B.2: Mechanical drawing (courtesy of the UCT Mechatronics Lab) of the bottom plate of the force platform

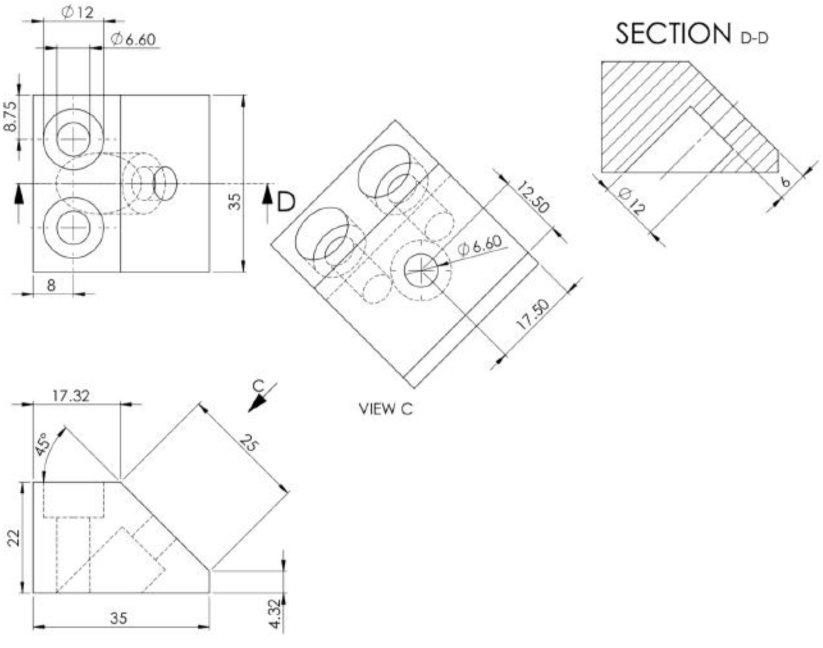


Figure B.3: Mechanical drawing (courtesy of the UCT Mechatronics Lab) of the load cell mounts used to secure load cells at 45°between bottom and top plates of the force platform

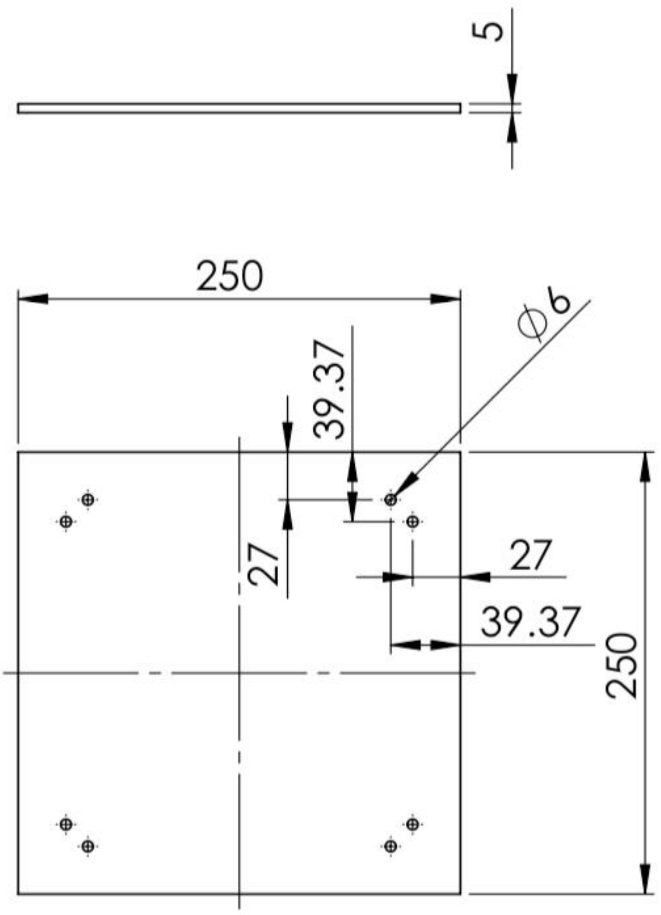


Figure B.4: Mechanical drawing (courtesy of the UCT Mechatronics Lab) of the top plate of the force platform

Appendix C

Embedded Software: Supporting Diagrams and Tables

C.1 STM32 Diagrams

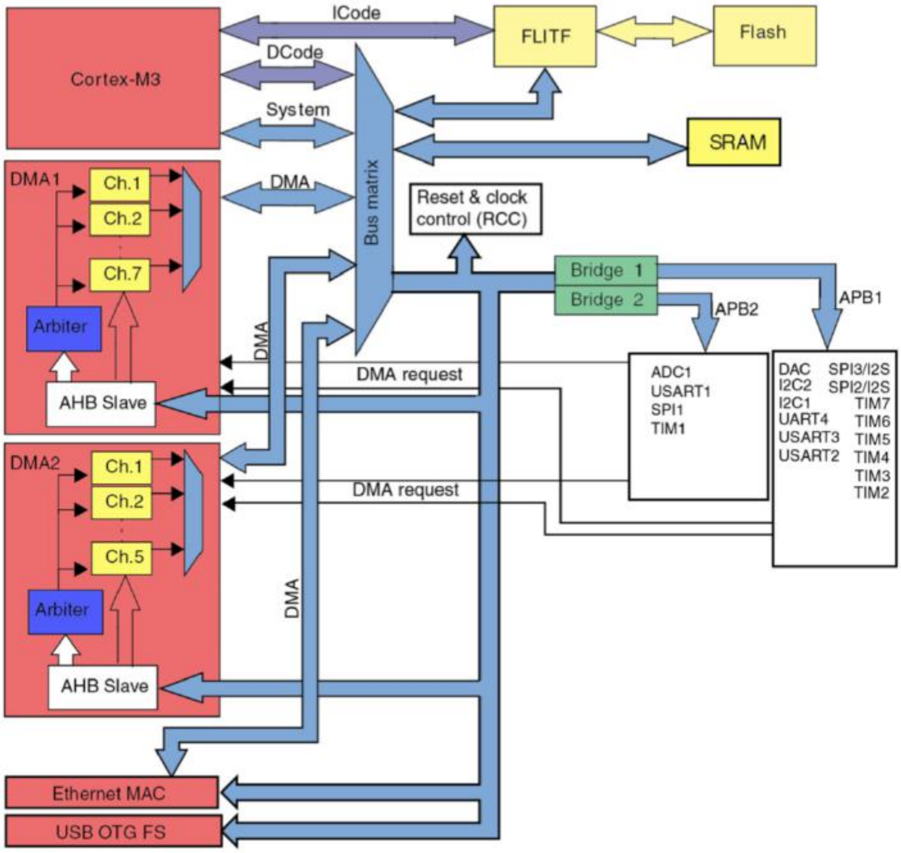
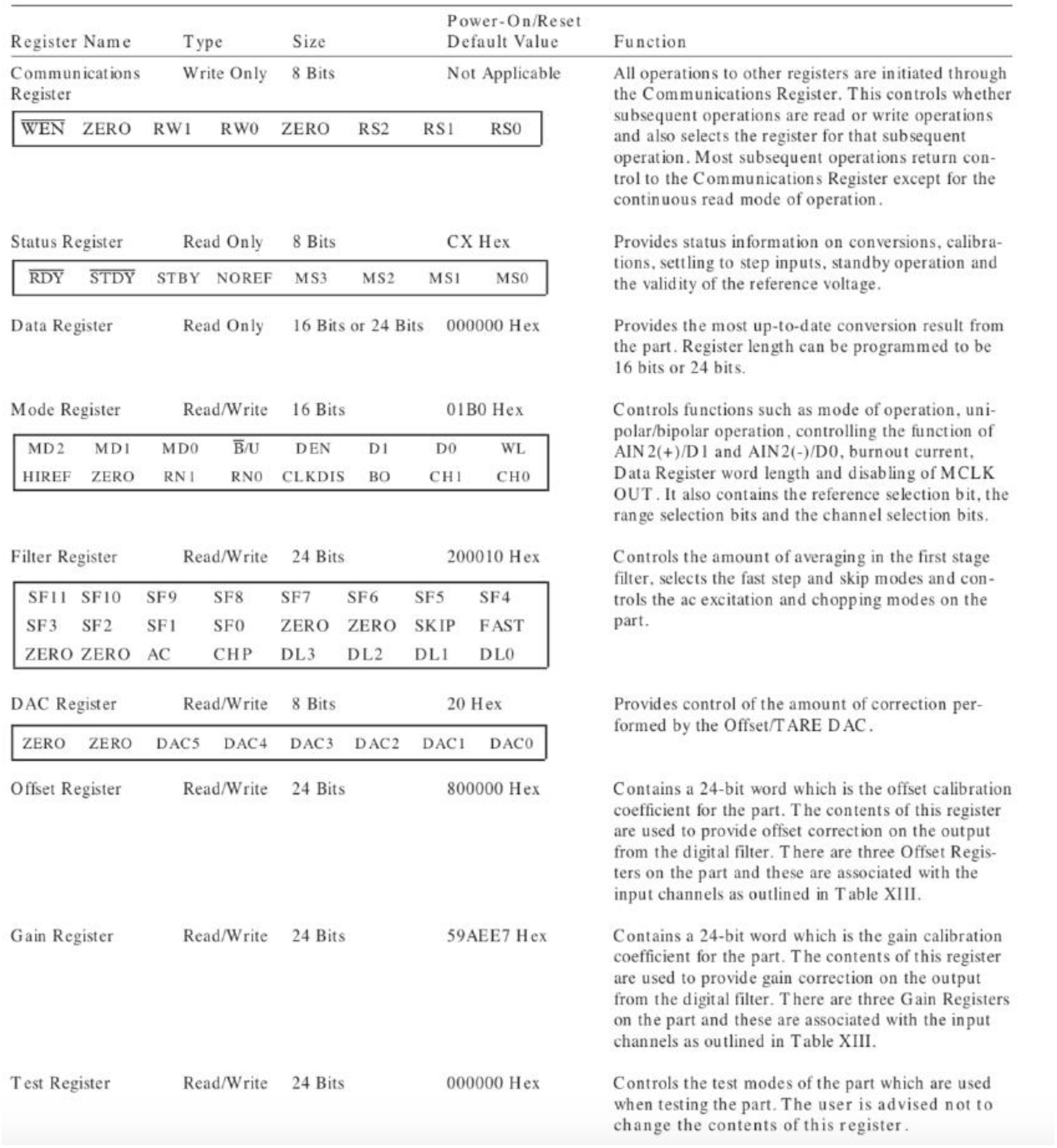


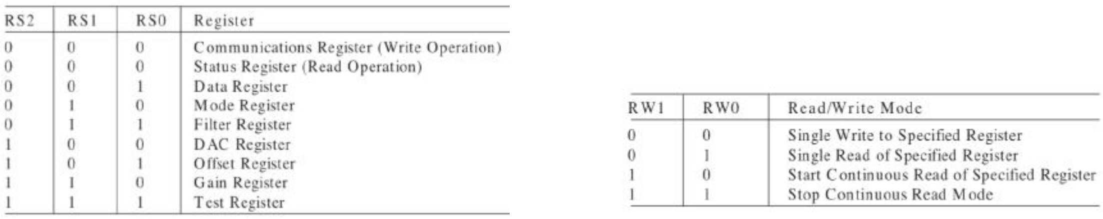
Figure C.1: Diagram showing the bus architecture of the STM32F1 from [52]. Notice DMA units with separate bus connections to the APB periherals. Note that the architecture for the STM32F4 is largely the same.

C.2 AD7730



Register Name	Type	Size	Power-On/Reset Default Value	Function
Communications Register $\overline{\text{WEN}}$ ZERO RW1 RW0 ZERO RS2 RS1 RS0	Write Only	8 Bits	Not Applicable	All operations to other registers are initiated through the Communications Register. This controls whether subsequent operations are read or write operations and also selects the register for that subsequent operation. Most subsequent operations return control to the Communications Register except for the continuous read mode of operation.
Status Register $\overline{\text{RDY}}$ $\overline{\text{STDY}}$ STBY NOREF MS3 MS2 MS1 MS0	Read Only	8 Bits	CX Hex	Provides status information on conversions, calibrations, settling to step inputs, standby operation and the validity of the reference voltage.
Data Register	Read Only	16 Bits or 24 Bits	000000 Hex	Provides the most up-to-date conversion result from the part. Register length can be programmed to be 16 bits or 24 bits.
Mode Register MD2 MD1 MD0 $\overline{\text{B}}$/U DEN D1 D0 WL HIREF ZERO RN1 RN0 CLKDIS BO CH1 CH0	Read/Write	16 Bits	01B0 Hex	Controls functions such as mode of operation, unipolar/bipolar operation, controlling the function of AIN2(+)/D1 and AIN2(-)/D0, burnout current, Data Register word length and disabling of MCLK OUT. It also contains the reference selection bit, the range selection bits and the channel selection bits.
Filter Register SF11 SF10 SF9 SF8 SF7 SF6 SF5 SF4 SF3 SF2 SF1 SF0 ZERO ZERO SKIP FAST ZERO ZERO AC CHP DL3 DL2 DL1 DL0	Read/Write	24 Bits	200010 Hex	Controls the amount of averaging in the first stage filter, selects the fast step and skip modes and controls the ac excitation and chopping modes on the part.
DAC Register ZERO ZERO DAC5 DAC4 DAC3 DAC2 DAC1 DAC0	Read/Write	8 Bits	20 Hex	Provides control of the amount of correction performed by the Offset/TARE DAC.
Offset Register	Read/Write	24 Bits	800000 Hex	Contains a 24-bit word which is the offset calibration coefficient for the part. The contents of this register are used to provide offset correction on the output from the digital filter. There are three Offset Registers on the part and these are associated with the input channels as outlined in Table XIII.
Gain Register	Read/Write	24 Bits	59AEE7 Hex	Contains a 24-bit word which is the gain calibration coefficient for the part. The contents of this register are used to provide gain correction on the output from the digital filter. There are three Gain Registers on the part and these are associated with the input channels as outlined in Table XIII.
Test Register	Read/Write	24 Bits	000000 Hex	Controls the test modes of the part which are used when testing the part. The user is advised not to change the contents of this register.

Figure C.2: Table showing a summary of the on-chip registers in the AD7730



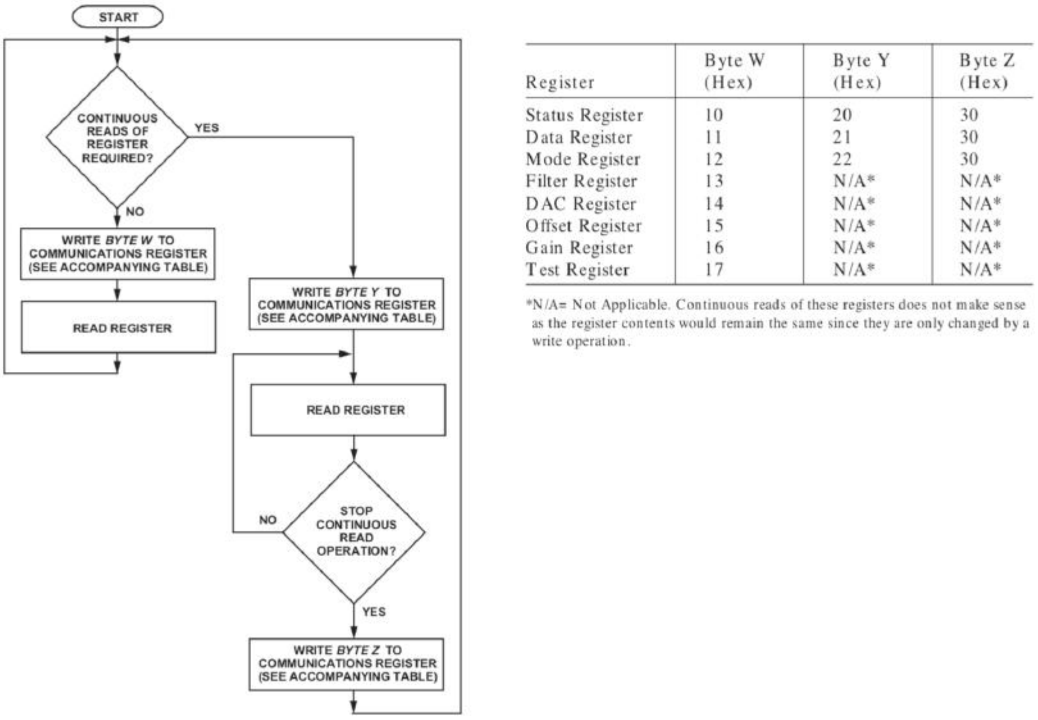
RS2	RS1	RS0	Register
0	0	0	Communications Register (Write Operation)
0	0	0	Status Register (Read Operation)
0	0	1	Data Register
0	1	0	Mode Register
0	1	1	Filter Register
1	0	0	DAC Register
1	0	1	Offset Register
1	1	0	Gain Register
1	1	1	Test Register

(a) Register select bits

RW1	RW0	Read/Write Mode
0	0	Single Write to Specified Register
0	1	Single Read of Specified Register
1	0	Start Continuous Read of Specified Register
1	1	Stop Continuous Read Mode

(b) Read/write bits

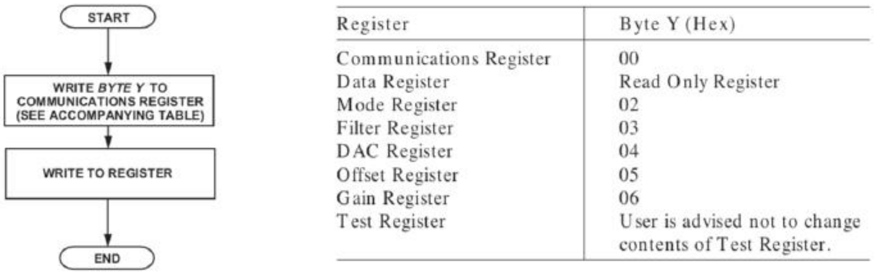
Figure C.3: Tables showing the bit patterns used to enable read/write mode as well as to select the target register for writing to/reading from


START
CONTINUOUS READS OF REGISTER REQUIRED?
YES
NO
WRITE BYTE W TO COMMUNICATIONS REGISTER (SEE ACCOMPANYING TABLE)
READ REGISTER
WRITE BYTE Y TO COMMUNICATIONS REGISTER (SEE ACCOMPANYING TABLE)
READ REGISTER
NO
STOP CONTINUOUS READ OPERATION?
YES
WRITE BYTE Z TO COMMUNICATIONS REGISTER (SEE ACCOMPANYING TABLE)

Register	Byte W (Hex)	Byte Y (Hex)	Byte Z (Hex)
Status Register	10	20	30
Data Register	11	21	30
Mode Register	12	22	30
Filter Register	13	N/A*	N/A*
DAC Register	14	N/A*	N/A*
Offset Register	15	N/A*	N/A*
Gain Register	16	N/A*	N/A*
Test Register	17	N/A*	N/A*

*N/A= Not Applicable. Continuous reads of these registers does not make sense as the register contents would remain the same since they are only changed by a write operation.

Figure C.4: Flowchart from [49] showing the procedure to read from the AD7730 registers in both single and continuous mode. A table of byte values used in the flow chart that correspond to the different target registers is also shown.


START
WRITE BYTE Y TO COMMUNICATIONS REGISTER (SEE ACCOMPANYING TABLE)
WRITE TO REGISTER
END

Register	Byte Y (Hex)
Communications Register	00
Data Register	Read Only Register
Mode Register	02
Filter Register	03
DAC Register	04
Offset Register	05
Gain Register	06
Test Register	User is advised not to change contents of Test Register.

Figure C.5: Flowchart from [49] showing the procedure to write to the AD7730 registers. A table of byte values used in the flow chart that correspond to the different target registers is also shown.

Appendix D

Literature Review Supporting Data

Data pertaining to the morphology of the cheetahs and greyhounds discussed in [22] is presented in Table D.1 below.

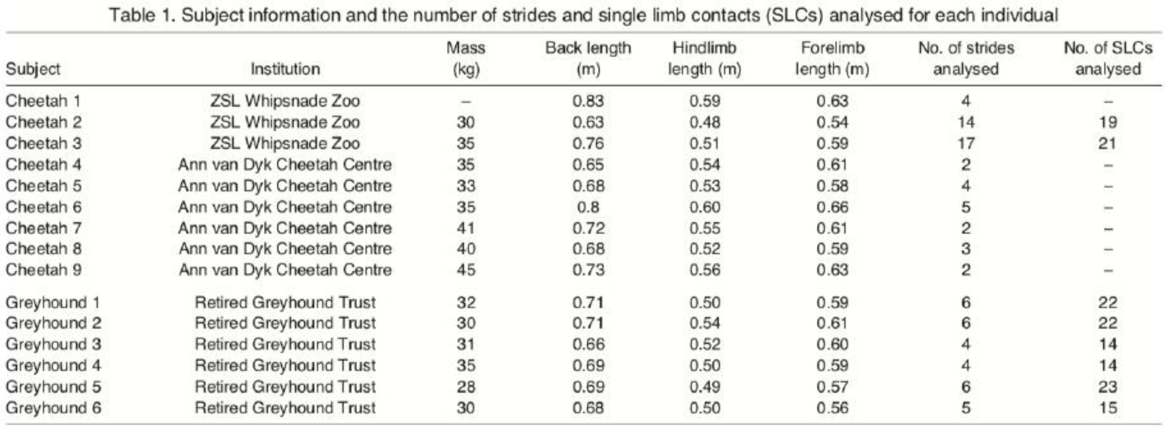


Table 1. Subject information and the number of strides and single limb contacts (SLCs) analysed for each individual

Subject	Institution	Mass (kg)	Back length (m)	Hindlimb length (m)	Forelimb length (m)	No. of strides analysed	No. of SLCs analysed
Cheetah 1	ZSL Whipsnade Zoo	–	0.83	0.59	0.63	4	–
Cheetah 2	ZSL Whipsnade Zoo	30	0.63	0.48	0.54	14	19
Cheetah 3	ZSL Whipsnade Zoo	35	0.76	0.51	0.59	17	21
Cheetah 4	Ann van Dyk Cheetah Centre	35	0.65	0.54	0.61	2	–
Cheetah 5	Ann van Dyk Cheetah Centre	33	0.68	0.53	0.58	4	–
Cheetah 6	Ann van Dyk Cheetah Centre	35	0.8	0.60	0.66	5	–
Cheetah 7	Ann van Dyk Cheetah Centre	41	0.72	0.55	0.61	2	–
Cheetah 8	Ann van Dyk Cheetah Centre	40	0.68	0.52	0.59	3	–
Cheetah 9	Ann van Dyk Cheetah Centre	45	0.73	0.56	0.63	2	–
Greyhound 1	Retired Greyhound Trust	32	0.71	0.50	0.59	6	22
Greyhound 2	Retired Greyhound Trust	30	0.71	0.54	0.61	6	22
Greyhound 3	Retired Greyhound Trust	31	0.66	0.52	0.60	4	14
Greyhound 4	Retired Greyhound Trust	35	0.69	0.50	0.59	4	14
Greyhound 5	Retired Greyhound Trust	28	0.69	0.49	0.57	6	23
Greyhound 6	Retired Greyhound Trust	30	0.68	0.50	0.56	5	15

Figure D.1: Table showing the mass and other physical characteristics of cheetahs and greyhounds observed in the study conducted in [22]

D.1 Commercial Force Plate Pricing

A quotation received from HiTech Therapy is presented below. The prices of three different popular commercial force plates are listed.

HiTech Therapy CC
50 Design Boulevard
Northlands Deco Park, New Market Road
Northriding GAUTENG 2010
Ph: 011-704 0002 **Fax:** 011-704 4999
Email: accounts@htherapy.co.za
Reg: CK94/007769/23 **VAT:** 4060142249

Quote

Number:	QU-CP1-000066326
Date:	10/16/2018
Page:	1
Reference:	
Expiry:	11/13/2018
Account:	UNIV17

30 Days From Statement

Sold To:
UNIVERSITY OF CAPE TOWN
UNIVERSITY OF CAPE TOWN
FINANACE
DEPARTMENT/CREDITORS SECTION
PRIVATE BAG X 3

Ship To:
CLINICAL RESEARCH
E-FLOOR E51 ROOM 51.44
OLD MAIN BUILDING, GROOTE
SCHUUR
OBSERVATORY

Customer VAT: 4540125707

Sales Person: JUAN PIENAAR

	Item	Description	Quantity	Unit	Unit Price	Discount	VAT	Total
1	9999	ACP - AccuPower Jumping and Power Analysis Platform ACP-A	1	EACH	280,000.00		36,521.74	280,000.00
2	9999	BP12001200 - Strain Gage Force Platform	1	EACH	940,000.00		122,608.70	940,000.00
3	9999	BP400600-OP - Force Platform BP400600-OP-1000	1	EACH	300,000.00		39,130.44	300,000.00

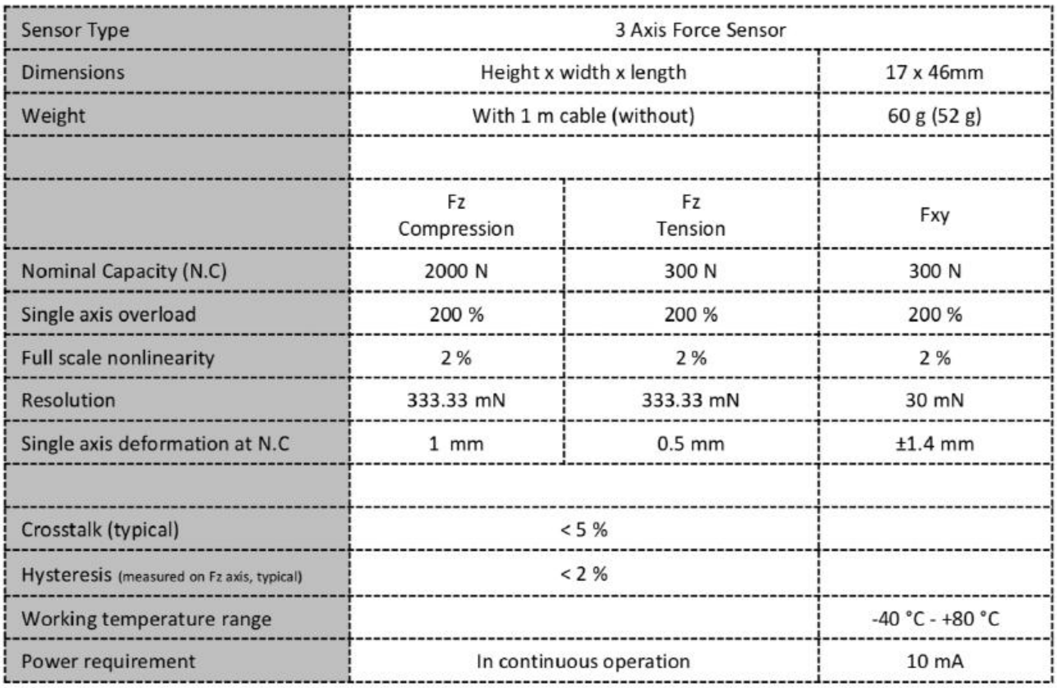
NOTE: PRICES SUBJECT TO EXCHANGE RATE FLUCTUATIONS.

BANKING DETAILS: STANDARD BANK SANDTON
BRANCH CODE: 019205 ACC NO: 022688307

Received in good order
Signed____________________ Date__________________

Subtotal:	R 1,321,739.12
Discount:	R 0.00
VAT:	R 198,260.88
Total:	R 1,520,000.00

D.2 OptoForce Specifications



Sensor Type	3 Axis Force Sensor		
Dimensions	Height x width x length		17 x 46mm
Weight	With 1 m cable (without)		60 g (52 g)
	Fz Compression	Fz Tension	Fxy
Nominal Capacity (N.C)	2000 N	300 N	300 N
Single axis overload	200 %	200 %	200 %
Full scale nonlinearity	2 %	2 %	2 %
Resolution	333.33 mN	333.33 mN	30 mN
Single axis deformation at N.C	1 mm	0.5 mm	±1.4 mm
Crosstalk (typical)	< 5 %		
Hysteresis (measured on Fz axis, typical)	< 2 %		
Working temperature range			-40 °C - +80 °C
Power requirement	In continuous operation		10 mA

Figure D.2: Table showing the performance specifications of the OptoForce OMD-45-FH-2000N 3-axis force sensor

Appendix E

Additional Circuit Schematics

A circuit schematic of the motherboard designed in Chapter 6 is presented in Figure E.1 on the following page. Note that GPIO pin net references (such as PA1) refer to those of the STM32F407 MCU.

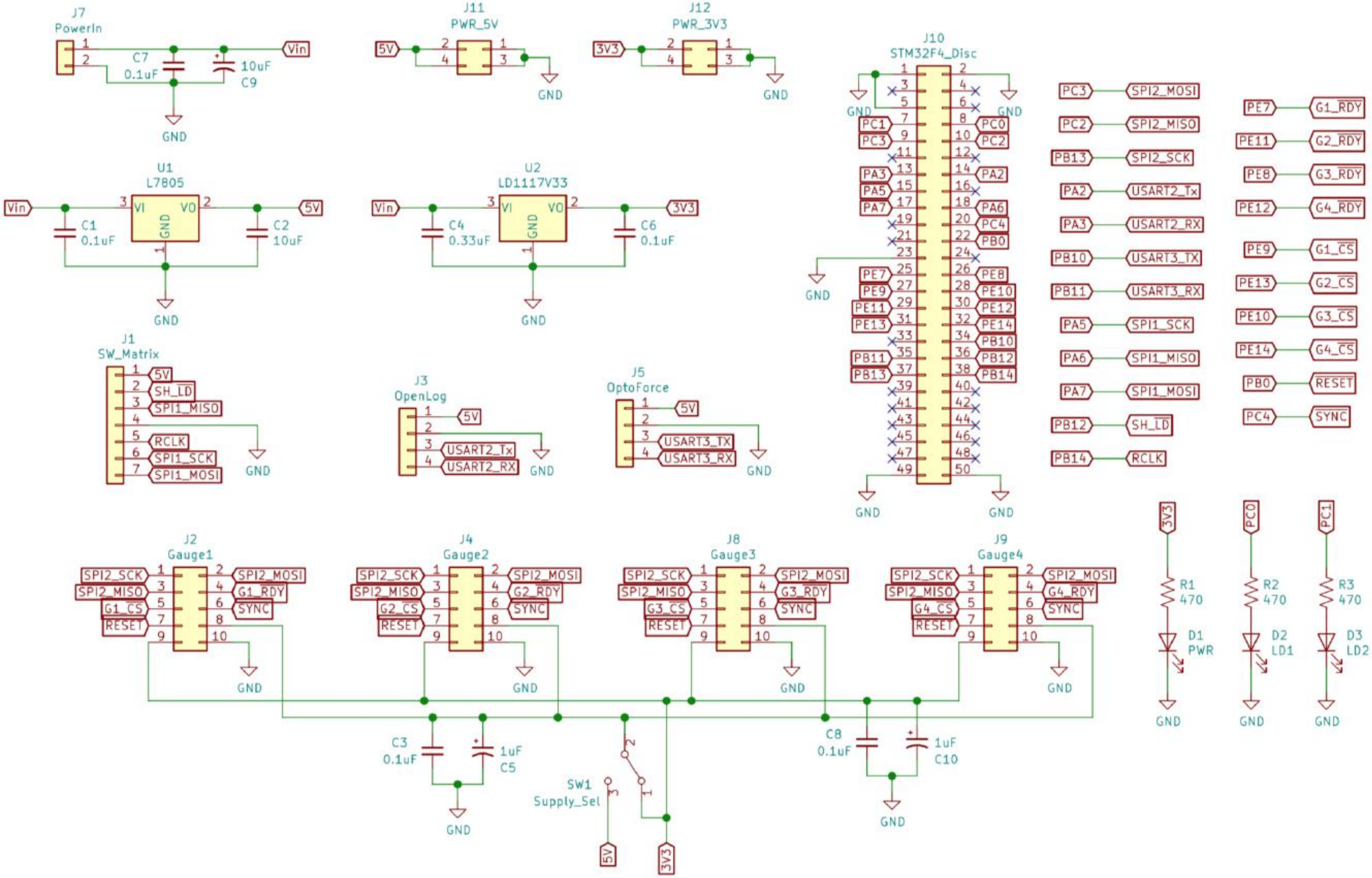


Figure E.1: Schematic diagram of the motherboard circuit